Automatic Voice Relay with Open Source Kiara

Long Yi and William D. Tucker
Department of Computer Science, University of the Western Cape, Bellville, South Africa
Telephone: +(27) 21 959-2461, Fax: +(27) 21 959-3006, Email: {**2475600**, btucker}@uwc.ac.za

Abstract—One way for Deaf people to communicate with hearing people over the telephone is to use a voice relay. The service is often provided with a human relay operator that relays text into voice, and vice versa, on behalf of the Deaf and hearing users. In developed countries, voice relay is frequently subsidised by governments or service providers. There is no such service in South Africa. We have built several automatic voice relay systems for a disadvantaged Deaf community in Cape Town. This paper describes how we augmented a general-purpose communication system for voice relay. Kiara is a fully open source Instant Messaging, voice and video over Internet Protocol communication system based on the Session Initiation Protocol. We integrated automatic speech recognition and text-to-speech technologies into Kiara to provide real-time automatic voice relay for relayed communication. As it stands, Kiara can also be used for standard voice and video relay with a human operator.

Index Terms—Deaf telephony, voice relay, open source, Session Initiation Protocol

I. Introduction

HEARING people use a variety of communication systems: voice telephony, video conferencing, Short Message Service (SMS), email, and Instant Messaging. Choosing an appropriate system to communicate depends on the context, as well as on the abilities and preferences of the user. Sometimes SMS is sufficient, whereas in other cases, voice telephony is needed to talk directly. Similar context choices are available for Deaf, hard of hearing and speech-impaired people, but their choices are limited by physiological constraints and socioeconomic situation, particularly in developing African countries like South Africa [1] [2]. Deaf users use sign language to communicate with one another, and may not be as text literate like other deaf people.

Users that cannot hear or use voice tend to use text telephony [3], whether text is appropriate or not, e.g. for personal or emergency needs. Note that poor text literacy can also impact negatively on using text telephony. Text telephones have been developed locally [4], and are common in the developed world [8]. Along with cell phones and PCs, such technologies provide an array of text-only telephony for Deaf users. Deaf people can also communicate with hearing people via some form of relay. A voice relay service (VRS) typically uses a human interpreter to translate a spoken language like English into text for the Deaf user. A video relay service uses video interfaces to relay sign language. Most of these relay systems rely on a human interpreter who translates the hearing user's spoken words into text or sign language for the Deaf user and vice versa. These services are often subsidised by governments and/or service providers. There is no such service in South Africa. Firstly, South African Deaf people are more likely to use SMS on a cell phone than a text telephone due to cost and other considerations [1][4]. Secondly, South Africa has lack of trained sign language interpreters, not to mention readily accessible video devices for the Deaf population. It is therefore understandably difficult and expensive to provide such a relay service. Thus, there is a need for a system that can provide automatic conversion to provide VRS without human interpreters. This is currently possible for voice relay.

A. Automatic voice relay

Figure 1 shows a scenario with automatic voice relay. When a hearing person calls a Deaf person, a voice conversation is initiated between the hearing participant and the Automatic Speech Recognition (ASR) service. During the conversation, the voice is sent to the ASR service and is interpreted into text. While keeping the voice connection open so that voice-to-text communication can continue, the ASR service translates the hearing person's words into text and sends it to the Deaf user. A textual version of the hearing person's spoken message appears on the display of the Deaf person's device, e.g. computer or cell phone. The Deaf person keys in a text response. The Text-to-Speech (TTS) Service receives the text and converts it to voice and forwards the voice to the hearing person's audio device.

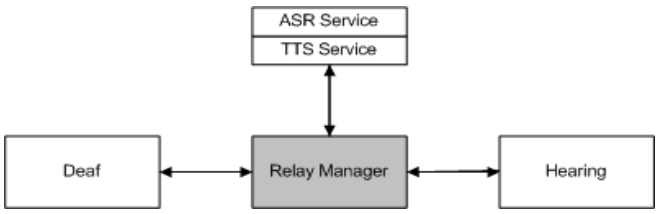

Figure 1 Voice relay converts text to voice to send to a hearing user, and then relays the hearing user's spoken message to text for a Deaf user.

B. Objectives

Third-party relay services that facilitate relayed communications for Deaf, hard of hearing and speech-impaired users are common in the developed world, e.g. Royal National Institute for the Deaf's Typetalk at British Telecom [5], TalkingText [6], and AT&T Relay Service [7]. These relay services are primarily operator-assisted, or manual. They make use of the Public Switch Telephone Network (PSTN), cellular networks and the Internet. Deaf devices are either text telephones or personal computers. These services are also provided free of charge to Deaf users in their respective countries. The primary goal of this project is to enable automatic relay between text and voice without operator assistance in order to keep costs down and remove dependence on a human interpreter.

To that end we have built several VRS systems for a disadvantaged Deaf community in Cape Town, South Africa [9][10][11]. The latest prototype was called SIMBA and was trialled extensively with Deaf users. Due to both technical and social considerations, especially lack of computer and text literacy, SIMBA did not achieve take-up in the Deaf community. However, its introduction nurtured an increase in both computer and text literacy that currently justifies the pursuit of an updated VRS system based on contemporary communication technologies. This paper describes a VRS called Kiara (daughter of SIMBA) that was designed for members of the Deaf Community of Cape Town (DCCT). Kiara is a completely free and open source (FOSS) software system that supports real-time communication of text, voice and video [12]. Kiara's basic text and video capabilities enable Deaf people to communicate with each other in text or sign language with video. Kiara can now also interchange text and voice communication to effect a VRS.

There are several obvious reasons to keep Kiara open source: open standards, customization, cost and the General Public License (GPL). Kiara is built on open standard protocols that make it able to connect to other software. Open source gives Kiara accessibility for other researchers and developers to add and customise features. Kiara is free and is therefore suitable for disadvantaged communities like DCCT. Kiara is developed on many open source libraries and most libraries are licensed under GPL.

This paper focuses on how Kiara integrates open source ASR and TTS technologies within the Session Initiation Protocol (SIP) architecture. Our approach is based on IETF RFC 3351 [14]. The next section provides a brief overview of Kiara's architecture. Section III discusses Kiara's Relay Manager that provides the automatic voice relay capabilities. Section IV discusses the implementation issues including the open source libraries and technologies used to build the voice relay system. Section V concludes the paper and Section VI suggests avenues for future work.

II. Overview of Kiara

Kiara is a SIP-based communication tool that supports synchronous communication of text, voice and video. Like Skype or Google Talk, Kiara is a voice over Internet Protocol (VoIP) application combined with Instant Messaging. Kiara's VoIP components are designed to work with Asterisk and SIP-capable IP-based communication systems and infrastructure. All Kiara components, on both client and server sides, are completely FOSS and standards-based. The Relay Manager provides voice relay services within the overall Kiara architecture (see). The rest of this paper concentrates on the Relay Manager's design and implementation.

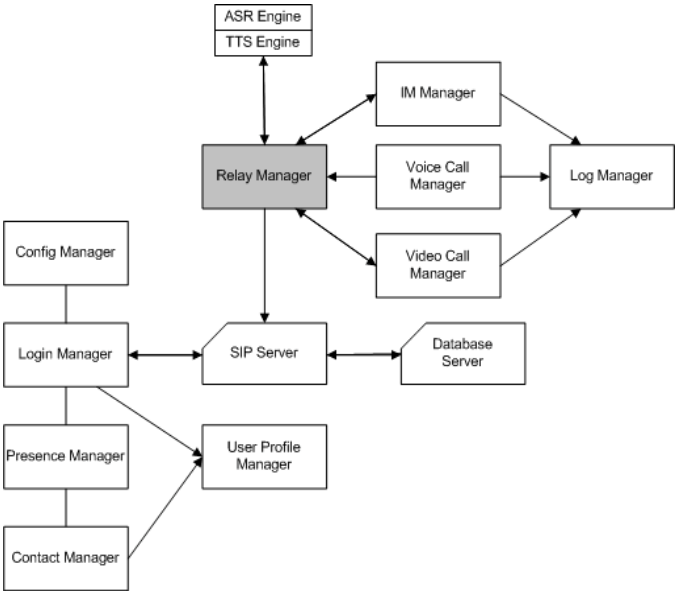

Figure 2 The Relay Manager is the Kiara component responsible for voice relay. The other components provide standard VoIP and IM functionality based on SIP.

III. Relay Manager Design Issues

Figure 3 shows the architecture of Kiara's Relay Manager. The design issues most relevant to relayed communication concern awareness during relayed conversation, the user interface and the communication flows from hearing user to Deaf user, and vice versa. This section describes each of these issues in turn.

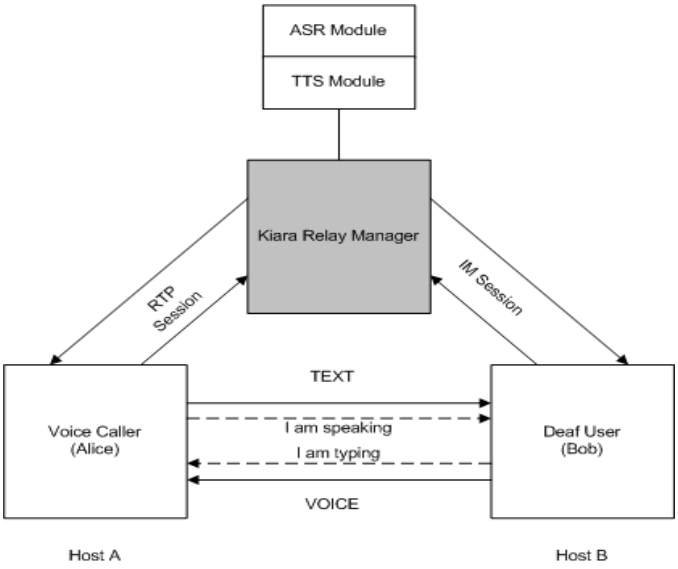

Figure 3 The architecture of Kiara's FOSS-based voice relay service closely resemble the voice relay scenario from Figure 1. The voice of user Alice is handled by SIP with Real-time Transport Protocol (RTP) [15]. The text to and from user Bob is handled by SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) [16].

A. Awareness in communication

Supporting awareness plays an important role in Kiara's communication flows, especially because ASR and TTS processing produce attendant latencies. A situation could easily arise where one Kiara user does not know that the other is typing or speaking. This results in the two users frequently typing or speaking either at the same time or in close succession. The worst situation is that one user might think the other is not responding when ASR or TTS processing takes a long time. These latencies and resulting expectations can disrupt the flow of the conversation. For example, Alice would hang up the call if she did not receive any notification that is Bob is either typing or having his text converted to speech for delivery. Thus, the Kiara clients support awareness by sending in-progress SIP messages, for both types of users. When Alice starts speaking, a notification message is created and transmitted to Bob. Bob receives a notification of the form "Alice is speaking". This provides a visual indication to Bob that Alice is speaking, thereby prompting Bob to wait until receiving processed ASR results. The Kiara client for the hearing user can also display an isTyping awareness notification. If the hearing user is not using a computer, Kiara can also send an aural isTyping notification to a VoIP or PSTN device. These awareness features aid the flow of relayed conversation by notifying end users of conversation flow.

B. User Interface

Due to the introduction of previous Deaf telephony prototypes, and repeated experience with computers at the computer lab since 2004, Deaf users at DCCT have a lot of experience with Instant Messaging, email and video chat. They are also familiar with Skype and Google Talk, so we used those tools as a reference to build the user interface to Kiara, shown in Figure 4.

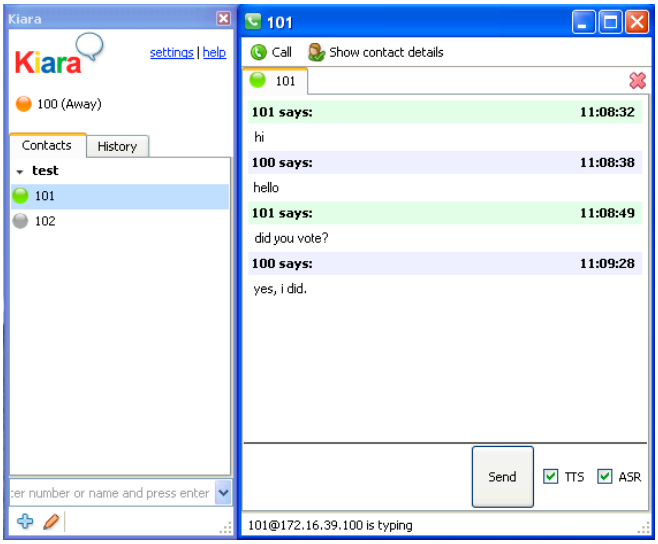

Figure 4 Kiara's user interface resembles other well-known Instant Messaging tools. A contact list (shown on the left) shows who is on or offline. A text messaging window timestamps messages and identifies who types what.

C. Communication flows

The flows of communication between the hearing user Alice and Deaf user Bob shown in Figure 5 are as follows:

1) Communication flow from hearing user to Deaf user

- Alice at host A makes a voice call to Bob at host B.
- Alice waits for the response message from Bob.
- When Bob takes the call, the Kiara client sends a SIP INVITATION message to Kiara's Relay Manager.
- Once the Relay Manager receives the INVITATION message from Alice within the timeout, the Relay Manager creates an RTP session with SIP, and initiates an ASR module to interpret the voice stream from Alice. When Alice keeps a silence longer than a specified timeout, the ASR module that is listening Alice's speech over RTP connection considers it the end of a message and converts the speech to text using an ASR engine.
- Then the Relay Manager sends the ASR results to Bob as an Instant Message to Bob as soon as it gets the results from the ASR module. In addition, if using a computer, Alice also gets the ASR results on her screen to so she can inspect what was sent to Bob.
- While Alice is speaking, Kiara also sends awareness messages, e.g. "Alice is speaking", and notifying Bob that something is happening. This is similar to an isTyping message.

2) Communication flow from Deaf user to hearing user

- When Bob sends a message to Alice, the text message is sent to Kiara Relay Manager first via an Instant Messaging channel, e.g. with SIMPLE.
- After receiving a text message from Bob, the Kiara Relay Manager initiates a TTS module to accept the text message from Bob. When Bob terminates a text message with a carriage return, the TTS module treats it as the end of a message and converts the text to voice using the TTS module.
- The synthesized speech is sent back to Alice over the existing RTP session.
- When Bob types, the Kiara client will also send aural awareness messages to notify Alice that Bob is typing.

IV. Implementation Issues

This section lists the FOSS technologies and protocols used to build Kiara's VRS and then shows how these technologies and protocols were implemented to create the service.

A. Technologies used in Kiara

1) SIP

SIP is a standard for establishing sessions over Internet Protocol (IP) [13]. SIP's capacity to simplify creation, management and termination of sessions is ideally suited to build communication systems for Deaf users [14]. SIP meets requirements for accessing a wide array of devices (telephone, mobile phone, PC) over a variety of networks (IP, WiFi, PSTN, 3G). This widespread availability enables any SIP-compatible device (soft-phone, telephone, SIP-enabled cell

phone, PBX, VoIP server) to work with our software. SIP is a text-based protocol. There are six SIP request messages and six SIP response messages. The Kiara client uses REGISTER message to login with the Kiara Relay Manager. INVITE and BYE messages establish and terminate a call session, respectively. Kiara Relay Manager uses a 200 OK message to accept a call session.

2) RTP/RTCP

RTP is a standard for the transport of real-time data, including audio and video [15]. RTP consists of data and control parts. The control part is also called Real-time Transport Control Protocol (RTCP). While RTP carries the media streams, RTCP carries the transmission statistics and quality of service information. When starting a voice call, a Kiara client sends a SIP INVITE message to the Kiara Relay Manager. The SIP message includes a Session Description Protocol (SDP) message that instructs the Relay Manager to open an RTP connection for the call. An RTP session is established between the Kiara client and Relay Manager. Voice streams are transferred over the RTP session.

3) SIMPLE

The Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE) is a standard for Instant Messaging based on SIP [16]. Kiara makes use of the SIP MESSAGE method as defined in RFC 3428 to support Instant Messaging and the SIP message SUBSCRIBE/NOTIFY for presence [16]. RFC 3856 defines two models: an end-to-end model and a centralized model [17]. Kiara uses a SIP server as a presence server to support the centralized model. This server handles all subscriptions. The SIP message PUBLISH allows Kiara Clients to inform the presence server about their subscription states [18].

4) XML

The Extensible Markup Language (XML) is a standard for creating and storing structured data. Kiara uses XML to store information like the contact list, user profile, and the exchange text messages.

5) Sphinx ASR

Sphinx is an open source project developed by the Sphinx Group at Carnegie Mellon University [19]. Sphinx 2 is a real-time, large vocabulary, speaker independent speech recognition system. It supports n-grams and finite state grammars (FSG) language models.

6) Festival TTS

Festival TTS is an open source text-to-speech engine developed by the Centre for Speech Technology at the University of Edinburgh [20]. It offers a general framework for building speech synthesis systems and synthesizes text into speech.

7) G.711/Speex

G.711 is a standard audio codec for modern digital telephone networks. Speex is a variable bitrate audio codec. It is able to dynamically modify its bitrates to respond to different network conditions. Speex is suitable for VoIP applications. Kiara uses both G.711 and Speex for real-time audio streams.

B. Implementation of Kiara

1) ASR Module

The ASR module of Kiara's Relay Manager is implemented using Sphinx2 [19]. It receives a voice stream from the Relay Manager, detects silence, converts voice to text and then sends the ASR results back to the Relay Manager. The ASR module can recognize digits very well (90%) is not as competent with words/sentences (40%). This is because the sample language model it provides was trained for American accents. There is now a tool called SphinxTrain that has been used for training of acoustic models and can train for South African accents. We will explore and test SphinxTrain in the future work.

2) TTS Module

The TTS module of Kiara Relay Manager is implemented using Festival TTS [20]. Festival receives text from the Relay Manager, converts text to voice and then sends TTS results back to the Relay Manager. At this time, Festival TTS translation only provides a male voice.

3) Relay Manager

The Relay Manager sends communication streams to a given Kiara client as soon as it receives the ASR/TTS results from the ASR/TTS modules. The Relay Manager is a small modified SIP User Agent (UA). It runs in the background to initialize the ASR and TTS modules, detects non-talk segments, creates RTP sessions, and transfers text/voice streams that are essentially ASR/TTS results to respective callers.

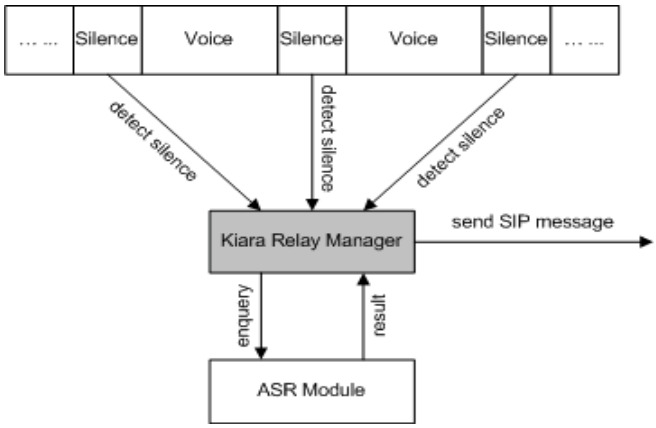

Figure 5 Silence and voice detection with attendant ASR processing.

As shown in Figure 5, “Voice” is defined by a contiguous series of sound data. A period of silence, e.g. 5 seconds, is defined as “Silence”. The “Voice” will be sent to Sphinx 2 ASR engine to convert voice to text, and the Kiara Relay Manager then relays resultant text. A “Silence” shorter than five seconds is regarded as a part of a speech and included in the “Voice”. A “Silence” longer than five seconds is discarded and the Kiara Relay Manager calls the ASR module to convert the preceding “Voice” segment. The “Voice” is currently

limited to a maximum length of fifteen seconds due to the limitations of Sphinx2.

Compared with detecting “Silence" from the audio client in Figure 5, detecting a text message is much easier (see Figure 6). If the Kiara Relay Manager receives a SIP Instant Message, it exacts the SIP message and sends the plain text to the TTS module for speech synthesis. The Kiara Relay Manager is also responsible for sending out the ASR results to the text client.

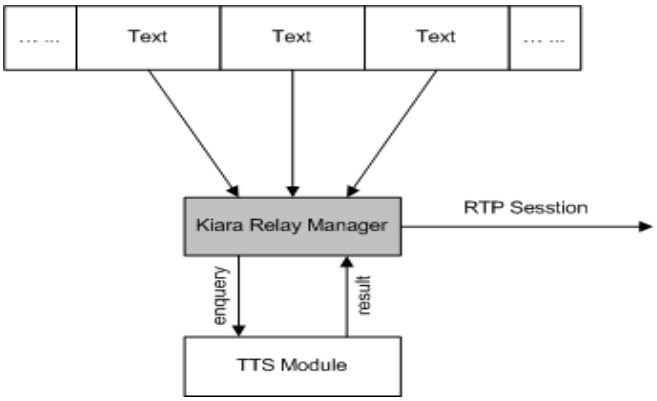

Figure 6 TTS processing is easier than silence detection because the SIMPLE messages already contain the text messages.

4) Awareness in Kiara Client

Kiara clients monitor keyboard activity and the audio input device. If input from the keyboard or audio device occurs within a particular time interval, a SIP message such as “KIARA_AWARE(X is typing)” or “KIARA_AWARE(X is speaking)” is created and delivered by the Kiara Client. The SIP message looks like this:

```
MESSAGE sip:bob@softbridge.uwc.ac.za SIP/2.0
Via: SIP/2.0/TCP
alice.softbridge.uwc.ac.za;branch=y8rD5cG655xdgcfk
Max-Forwards: 70
From: sip:alice@softbridge.uwc.ac.za;tag=49583
To: sip:bob@softbridge.uwc.ac.za
Call-ID: bcb66bgc55r@172.16.39.5
CSeq: 1 MESSAGE
Content-Type: text/plain
Content-Length: 18

Alice is speaking.
```

Actually, the text of the SIP message is coded in XML like this:

```
<sipmessage>
    <message type="AWARENESS">
    <fontface>MS Sans Serif</fontface>
    <fontcolor>#000000</fontcolor>
    <text>Alice is typing</text>
    </message>
</sipmessage >
```

Note that “message type” above is a flag to distinguish a SIP Instant Message from a SIP awareness message. When Kiara receives a message, it decodes and extracts the XML encoded message. If the message type is an Instant Message, the message is displayed on the Deaf user’s interface. If the message type is an awareness message, the message is displayed as a notification on the interface.

5) Open source libraries

There are several open source libraries and packages that Kiara is built on. oSIP is an open source implementation of SIP and SIMPLE, written in C [21]. Kiara uses this library to provide SIP signalling, Instant Messaging, awareness and presence. oRTP is an open source implementation of RTP/RTCP. oRTP is also written in C [22]. Kiara uses this library to provide voice streaming. PortAudio is a cross-platform audio I/O library [23]. Kiara uses this library to capture and process voice streams. Kamailio (previously called OpenSER) is a mature, flexible and scalable open source SIP server [24]. It is configured as a SIP registrar, SIP proxy, location, and presence server to support SIP–based network infrastructure. TinyXML is a simple, small and stable XML parser that written in C++. It is easy to use and can be easily integrating into other programs [25]. Kiara uses TinyXML as a text file parser to create and save human readable data like text, awareness messages, contact information, and log information.

V. CONCLUSION

In conclusion, Kiara is a SIP-based communication tool that supports synchronous communication of text, voice and video. ASR and TTS technologies were integrated into Kiara's Relay Manager in order to support automatic voice relay for a Deaf users to communicate with a hearing user. The architecture of Kiara Relay Manager is based on [14]. All of the components are completely FOSS. Textual and audio awareness mechanisms help end users deal with the latencies incurred by relayed conversation. Kiara's appearance closely resembles common Instant Messaging interfaces and is therefore familiar to computer literate users. Kiara's standards-based design and implementation mean that Kiara can connect to any form of telephony device without modification. Note that even without the automatic VRS capability, Kiara can be used to provide a voice and/or video relay service just like any other such service available in developed regions. The only holdback for Deaf users in South Africa is that such services are not freely available.

VI. FUTURE WORK

We have not yet tested the automated Kiara VRS with actual users at DCCT. First, we wish to ensure that the system is robust enough to perform unattended. Second, we must identify real-life scenarios where Deaf and hearing people would want to use the system. There is no culture of relayed communication in South Africa. That was the main hindrance to take-up in prior systems. We are hoping that improved computer and English literacy will increase the chances of

Kiara's success.

Kiara's ASR and TTS modules currently only support English. Thus, Kiara still needs several extensions to support other South Africa languages. The acoustic model for Sphinx in the ASR module is designed for English. We may need to build an acoustic model and extend the ASR module to able to recognize Afrikaans and Xhosa, for example.

If we find that ASR performance is not acceptable for end users, we will experiment with a human relay operator. We may also use a human sign language interpreter to translate between sign language and voice with the hope that someday automated sign language recognition and generation will become as common (and open source) as recognition and generation of text.

ACKNOWLEDGMENTS

The authors thank the staff and members of Deaf Community of Cape Town (DCCT) for their participation in the project. We also thank Telkom, Cisco and THRIP for financial support via the Telkom Centre of Excellence (CoE) programme. This project is also financially supported by SANPAD, the South Africa-Netherlands Research Programme on Alternatives in Development.

REFERENCES

[1] Glaser M and Tucker WD (2004). Telecommunications bridging between Deaf and hearing users in South Africa. Proc. Conference and Workshop on Assistive Technologies for Vision and Hearing Impairment (CVHI 2004), Granada, Spain.

[2] Agboola I.O. and Lee A.C. (2000). Computer and Information Technology Access for Deaf Individuals in Developed and Developing Countries. Journal of Deaf Studies and Deaf Education, Oxford University Press, 5(3), 286-289.

[3] Power MR and Power D (2004). Everyone Here Speaks TXT: Deaf People Using SMS in Australia and the Rest of the World. *Journal of Deaf Studies and Deaf Education*, 9(3), 333-343.

[4] Glaser M (2000). A Field Trial and Evaluation of Telkom's Teldem Terminal in a Deaf Community in the Western Cape. *Proc. South African Telecommunication Networks and Applications Conference,* (*SATNAC 2000*), Stellenbosch, South Africa.

[5] Deaf's Typetalk at British Telecom, the UK's text to voice relay service, http://www.typetalk.org

[6] TalkingText, http://www.talkingtext.net

[7] AT&T Relay Service, the US's relay service, supports TTY, IP, Video and IM relay, http://www.consumer.att.com/relay

[8] Verlinden M. (2000). Computer Applications for Deaf information and communication, developed/under development at IvD/MTW (RDS-department),Proc. 19th International Congress on Education of the Deaf (iced 2000), Sydney, Australia, (CD-ROM publication).

[9] Penton J, Tucker WD and Glaser M (2002). Telgo323: An H.323 Bridge for Deaf Telephony. *Proc. South African Telecommunications Networks & Applications Conference,* (*SATNAC 2002*), Drakensberg, South Africa, 309-313.

[10] Tucker WD, Glaser M and Lewis J (2003). SoftBridge in Action: The First Deaf Telephony Pilot. *Proc. South African Telecommunications Networks & Applications Conference,* (*SATNAC 2003*), George, South Africa, II-293-294.

[11] Sun T and Tucker WD (2004). A SoftBridge with Carrier Grade Reliability Using JAIN SLEE. *Proc. South African Telecommunications Networks & Applications Conference,* (*SATNAC 2004*), Stellenbosch, South Africa, II-251-252.

[12] Yi L and Tucker WD (2008). Kiara: an open source SIP system to support Deaf telephony, Southern Africa Telecommunication Networks and Applications Conference (SATNAC), Durban, South Africa

[13] Rosenberg J., Schulzrinne H., Camarillo G., Johnston A., Peterson J., Sparks R., Handley M., and Schooler E. (2002). SIP: Session Initiation Protocol. RFC 3261, *IETF*.

[14] Charlton N., Millpark, Gasson M., et al. (2002), User Requirements for the Session Initiation Protocol (SIP), in Support of Deaf, Hard of Hearing and Speech-impaired Individuals, RFC 3351, *IETF*.

[15] Schulzrinne H., Casner S., et al. (1996) RTP: a transport protocol for real-time applications, Request for comment, RFC 1889, *IETF*.

[16] Campbell B, Rosenberg J, Schulzrinne H, Huitema C and Gurle D (2002). Session Initiation Protocol (SIP) Extension for Instant Messaging. RFC 3428, *IETF*.

[17] Rosenberg J. (2004) A Presence Event Package for the Session Initiation Protocol (SIP), RFC 3856, *IETF*.

[18] Niemi A., Ed. (2004) Session Initiation Protocol (SIP) Extension for Event State Publication, RFC 3903, *IETF*.

[19] The Sphinx Group at Carnegie Mellon University, Sphinx2, http://cmusphinx.sourceforge.net/html/cmusphinx.php

[20] The Center for Speech Technology at the University of Edinburgh, Festival, The Festival Speech Synthesis System, http://www.cstr.ed.ac.uk/projects/festival/

[21] Moizard A. (2002), The GNU oSIP library – a session initial session library (RFC3261), http://www.gnu.org/software/osip/

[22] Morla S. (2007), oRTP – a real-time transort protocol library (RFC3350), http://www.linphone.org/index.php/eng/code_review/ortp

[23] Bencina R., PortAudio – a portable cross-platform Audio API, http://www.portaudio.com/

[24] Mierla D.C., Modroiu E.R. (2005), Kamailio (OpenSER) - the Open Source SIP Server, http://www.kamailio.org/

[25] Lee T. (2002), TinyXML – a simple, small C++ XML parser, http://www.grinninglizard.com/tinyxml/

Long Yi holds a Masters degree in Computer Science from the University of the Western Cape (UWC). He is the lead technical programmer for the Broadband and Applications Networks Group (BANG) at UWC, primarily concerned with FOSS-based communication tools.

William D. Tucker is a senior lecturer in Computer Science at UWC and leads BANG research there. He recently completed a PhD on design and evaluation abstractions for information and communication technology for development. One of his field studies concerned Deaf telephony.