A Mobile Deaf-to-hearing Communication Aid for Medical Diagnosis

Muyowa Mutemwa 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: {**2550606**, btucker}@uwc.ac.za

Abstract— This paper describes how a deaf to hearing communication aid built for a mobile phone can be used to provide semi-synchronous communication between a Deaf person and a hearing person who cannot sign. Deaf people with access to mobile phones have become accustomed to using Short Messaging Services, to communicate with both hearing and Deaf people. However Most Deaf people have basic literacy levels and hence prefer not to communicate with text, but with South Africa Sign Language. The prototype uses interpreted communication between sign language and English. The mock-up is meant to help a Deaf person convey their medical conditions to a doctor face-to-face in the office. The prototype is made using pre-recorded sign language videos for the Deaf person and English text for the hearing doctor. The interaction on the mobile phone is done inside the phone's browser using video streaming, instead of playing the video in a third-party media player. The design goal was to present the system on a mobile phone from the computer-based prototype. This paper takes a look at the background, related systems, the methods, the design and user testing of such a system on a mobile phone; using two prototypes client-server and client only.

Index Terms—Network services, web service, mobile services, Deaf telephony

I. Introduction

This paper describes a communication aid on a mobile phone that helps Deaf users who can only use South African sign language (SASL) to communicate with a hearing doctor that cannot sign. Deaf with a capital 'D' is different from deaf or hard of hearing in that Deaf people primarily use sign language to communicate. SASL defines a sense of culture and identity much like the eleven official South African languages define the culture and identity of the groups that use spoken, and consequently textual, languages like isiXhosa and Afrikaans. Because Deaf people are not literate in spoken and written languages, they struggle with basic human interaction such as visiting a doctor for medical diagnosis [1]. This mobile phone mock-up is a potential solution to such a problem. The concept was designed and tested on a computer by a visiting Dutch MSc student in Industrial Designing [2]. This paper describes an implementation of the mock-up on the phone and it is important to first understand how that mock-up operates.

The mock-up, called SignSupport, runs on a computer in a browser, and each simulated mobile page displays a pre-recorded SASL video and English text. The SASL videos are questions and answers that help a Deaf person tell a doctor what is wrong. This mock-up is not an expert system, does not use artificial intelligence and it does not give a diagnosis. Rather, once the Deaf person has completed answering the questions shown in sign language, the mobile mock-up produces English text representing what the Deaf person has defined, and can be handed to the doctor to read what the problem is in plain English.

The hearing doctor can then respond to the Deaf person using a 'phrase book' that looks up English phrases and displays a sign language equivalent. This is not automatic text recognition; merely a simple domain-limited lookup of common phrases; in this case medical advice and/or directives for the Deaf person in sign language.

The mock-up requires only a mobile phone with a data connection running a browser that supports Small Web Format (SWF). The intention is to run the system *within* the mobile browser instead of using a third party media player to make it easier to develop and use [3]. The original SignSupport mock-up and the follow up implementation on a mobile phone, described in this paper, were informed and tested by Deaf users associated with the Deaf NGO, an NGO (non-governmental organisation). All participants mentioned in [2] and herein are Deaf NGO staff members. Because they work for the Deaf NGO, they have basic English literacy, computer and mobile literacy. It is notable that none of them have matriculated high school. Most interaction with participants was therefore conducted with the help of a SASL interpreter.

Interpretation between sign language and spoken/written language remains a challenge. A variety of interpretation systems have been built to ease communication between Deaf and hearing people: some manual and some automated, to varying degrees of success. Some of these systems are described in Section II. Unfortunately, even the most basic tele-services for Deaf people either were discontinued [http://www.dacst.gov.za/speeches/minister/mar2002/tissa_launch.htm], are too expensive [http://thibologa.co.za/mobile_serv.html#dictionary] or are in basic [[4], [5]], as opposed to applied, research stages. In the public space, there are some attempts to bridge the gap, for example, by including sign language interpretation during news bulletins on television, but very little on mobile phones which is strange because mobile phones represent Deaf people's only real remote communication option because of a phone's text and video capabilities coupled to its accessibility and, at least for text with Instant Messaging such as MXit, inexpensive.

The rest of this paper is as follows. Section II briefly covers work related to sign language communication technology and the medical diagnosis domain. Section III describes the methods used to build and evaluate prototypes of the communication aid. Section IV describes two mobile prototype: client-server and client-only. Section V reports on trials with Deaf users conducted at the Bastion. Finally, Section VI concludes the paper and indicates future work.

II. Related Work

A. Expert Systems

An expert system is software programmed using Artificial Intelligence (AI). These systems use huge libraries to phrase questions and based on the questions calculate the likelihood of a disease and give advice or give a medical diagnosis [6]. These libraries have been compiled with questions for symptoms relating to diseases. Expert systems have to evaluate all possible diseases listed in a data from the user input, making the system not practical and user friendly [7]. Some expert systems using AI have reached levels of performance that are equivalent to human experts [8]. Some expert systems use decision trees in automated acquiring knowledge from databases. The results of this approach have showed the method as appropriate. Small expert systems are normally judged as not easily extensible, but these can be made to create great accuracy in performance over its small space of possibility. These small systems have proven to be unreliable in diagnosing rare diseases.

According to Verlinden, expert systems can be built for patient side diagnosis that means a user can monitor his/her health status from home and that can help in early diagnosis of the disease [6]. Such systems can get the symptoms and laboratory results from the user and suggest a curing method after the diagnosis has been made. User authentication is important in establishing the identity of a user and creating a personal record, which can be stored in the databases. Using a limited number of questions in the system, allows for chances of overlooking the seriousness of the disease [8]. Users prefer an expert system that has limited questions and is user friendly. A good expert system should allow the user to enter the complaint with a minimum number of interactions. The success or failure of an expert system is determined by its accuracy and rate of diagnosis [7]. Further more for an expert system to be successful, it has to be simple and specific in its function. Expert systems are mostly made of circles that contain testing evolving programs and noting problem-solving strategies. So far the proof of the AI concept has been the main drive behind development of expert systems [8].

B. Sign Language-to-text systems

Sign language-to-text is a process that translates sign language such as American Sign Language (ASL), Arabic Sign Language (ArSL), etc into text. Prototypes that translate sign language into text have been invented to ease communication between Deaf and hearing people. A majority of these systems use the data/power gloves technique to help the system recognize the sign made by a person. Some of these systems use the hidden Markov model (HMM) to archive low error rates for both the training process and the testing process to obtain reasonable results. Testing on such system is done with input from a different set from the set it trained with [9].

Sign language-to-text examples can also be found in projects such as gestures recognition using feature vector [[4], [5]]. A feature vector is then used to describe a distinct sign signature that is generated. The feature vectors are then used to train the HMM with signed words and phrases. The interaction between the database and the HMM which produced the highest probability is then used to choose the most possible interpretation of the sign [[4], [5]].

In England they have good relay systems such as TESSA* [10], where public workers can use the system to communicate with Deaf people. Currently in SA, there is a service that can convert text to a sign language video (one direction from a hearing person to a Deaf person) using SMS and Multimedia Messaging Services (MMS), but this is costly [http://thibologa.co.za/mobile_serv.html#dictionary].

C. Other sign language relay systems

Most Deaf people use services like short messaging services (SMS), Telephone Typewriters (TTY), email, chat Instant Messengers (such as Google talk, mxit, etc), fax, video telephony and voice/TTY relay services to communicate with each other or with hearing people [11]. In some countries there are call centers that are accessible for Deaf people. While hearing people can speak to emergency call operators with ease, Deaf and hard of hearing people have problems in doing so. Most Deaf people are moving towards internet-based technologies, SMSs, and IM applications [12]. It would be wise to use such technologies to help better the communication between a Deaf person and hearing person. In SA advanced mobile phones support video calling, but the poor resolution and low frame rate of the video remains unacceptable for SASL communication. Voice transport in the video calling is still prioritized but this is not needed for sign language communication.

III. Methods

Achievements made by expert systems to date are modest, they hold great promise and attract a wide spread of interest. However, expert systems, sign language-to-text and other systems that use AI find it hard when it comes to general problem solving, choosing the procedure to use, analysing reliability of facts and solutions [8]. Deaf people at the Deaf NGO cannot communicate with voice. Even though they can use SMS and email like hearing people, they prefer to communicate in sign language and not text.

The mobile communication aid described below is not an expert system because it does not use AI to give its user a diagnosis but makes use of an actual doctor. The system sums up in text shown on the mobile device (see Figure 7.b), the symptoms the Deaf person is feeling for the hearing doctor to diagnose; and provides a phrase book mechanism for the doctor to ask questions of the Deaf user with sign language videos, (see Figure 1) for the system design outline.

The prototype is designed to fit mobile phone screens. As long as the video and the text can be seen clearly (Figure 1), the design of the prototype fits only selected phones. The prototype was evaluated with Deaf NGO staff members (see Section V). The mobile phones mentioned here are able to browse the Internet using Wireless Fidelity (Wi-Fi), have a browser that play embedded video within the browser itself, and have more than 1 gigabyte of storage space. Most interaction with participants was therefore conducted with a SASL interpreter.

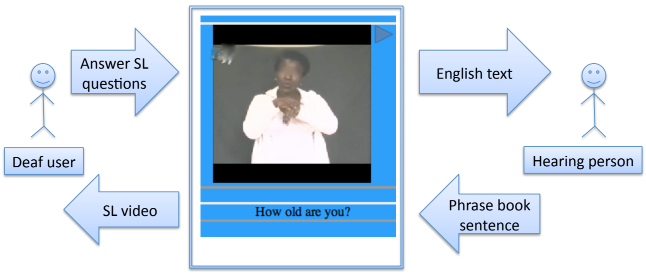

Figure 1: The high-level design of the system. The Deaf user inputs answers from sign language video questions and the hearing user reads English text. The hearing user inputs an option of a sentence in the dictionary and the Deaf user sees a sign language video.

A. Users Interaction Design

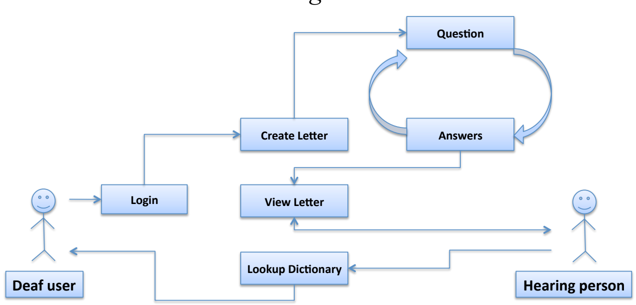

Figure 2: How the Deaf and hearing users interact with the system. The Deaf user creates English text by answering questions while the hearing person read the letter and responds using a lookup dictionary.

The Deaf user logs on (see Figure 2); navigates through a series of web pages with questions or answers. Each web page has a sign language video of a question or answer and English text. At each stage the user watches a sign language question on one page, proceed to the next set of pages that have sign language video answers. This process is repeated until the questions are answered. The questions do not follow each other linearly but have a tree like structure; the design of this structure cannot be shown in this paper due to page limitation.

Once the user has completed answering the questions, they can show the hearing user the text on the system. The hearing user then reads the text and responds to the Deaf person using the lookup dictionary. The lookup dictionary contains relevant words and instruction on how to take the medication.

B. Technologies used in prototype

Hypertext Preprocessor (PHP) is used mainly as a scripting language and embedded into the HTML source document to produce dynamic pages. A web server with a processor module interprets PHP source documents to produce the web page document [http://wiki.php.net/]

Extensible Hypertext Markup Language (XHTML) is a combination of HyperText Markup Language (HTML) and Extensible Markup Language (XML) a form of a language that allows you to define tags, while having the qualities of a HTML [http://www.w3.org/XML/].

Small Web Format (SWF) is a partial open repository for multimedia and vector graphics created by Adobe. Intended to be small enough for publication on the web, it contains animations or applets of varying degrees of interactivity and function. SWF can only run in mobile phones that are Adobe Flash enabled, these phones then make use of either Adobe Flash Player or the Adobe Integrated Runtime to play the SWF files. [http://www.adobe.com/devnet/swf/].

JavaScript is an object-oriented scripting language that enables web pages to access computational objects within an environment. It is best used in the form of client-side JavaScript standard [http://www.ecmascript.org/].

Personal Apache, MySQL and PHP (PAMP) is a Nokia S60 version of the well known LAMP, it used to create public dynamic web pages on the Nokia phones running Symbian Operating System (OS). XML was also used [http://php.net/].

The E71 and N82 are Smart phones from the Nokia Eseries range with a QWERTY keyboard and the Nokia Nseries with the N-Gage gaming platform. Both phones run on Symbian OS v9.2 and belong to the Series 60 3rd Edition. The current web browser on the phones supports Wireless Access Protocol (WAP) 2.0, XHTML, HTML and Wi-Fi 802.11 standard. It has 128RAM for PAMP processing and can support up to 16GB for sign language video storage [[http://europe.nokia.com/find-products/devices/nokia-e71/specifications], [http://www.forum.nokia.com/devices/N82]].

C. Content creation design for the mobile-based prototype

The user interface was created using PHP. The system administrator can login, create questions and answers, the hearing person's view, and add words to the phrase book, making the system user friendly (see Figure 2).

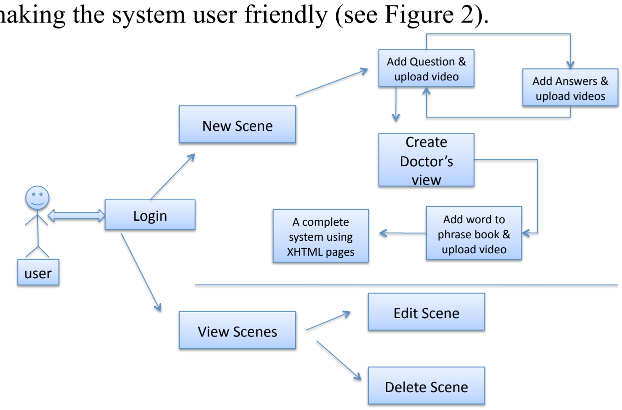

Figure 3: High-level Design of how the system supports content independent creation. The user is an Administrator of the server, who records videos and creates content for a doctor's visit. After the new content is created it is then uploaded on the server (see section IV.A) or on the client (see section IV.B).

The system is designed to support content independency such as a doctor's visit, buying medication at a pharmacy, ticket purchasing at a train-station, etc. These can be created with ease by just uploading a sign language video and writing English text (see Figure 3).

Each page on the system is stored in a XHTML document, because mobile phone browsers except this format and display it easily. Video streaming in the mobile phone browser can only happen if the video is encoded into the SWF format, all the sign language videos had to be converted into this format.

This is done to avoid playing the videos with a third-party media player like RealPlayer on Symbian, but rather in the browser it self.

Once the system is created it is meant to run on a mobile phone. The system can then either be stored on a server running on a mobile phone using PAMP or on a computer using Apache, MySQL and PHP. The prototype runs on a mobile phone but was implemented in two ways, the first prototype's server was a mobile phone and the second's server was a computer.

IV. TWO MOBILE-BASED PROTOTYPES

A. *Client-Server prototype*

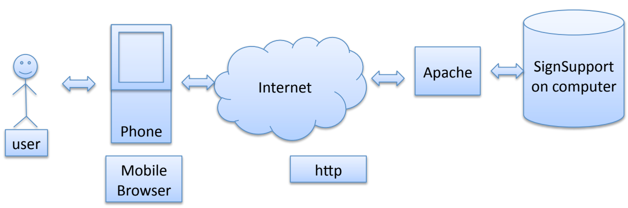

Figure 4: High-level design of system stored on computer and accessed in the mobile phone browser using the Internet. The user still uses a mobile phone browser but the contents of the system are stored online.

The system content is stored on the PC and accessed on the phone (see Figure 4). The XHTML pages are divided into sections to allow refreshing of contents on the page, each refreshing of the browser page displays new contents through IP from the SignSupport system on the computer (see Figure 4). The videos and the text for each question or answer are stored in an XML file and as the user navigates the XHTML pages, the JavaScript refreshes the page and displays new content on the page. User input and answers to the questions are then stored into the mobile phone's browser using cookies. The network connection used to exchange data between the computer and the mobile phone is a Wireless Local Area Network (WLAN).

B. *Server on mobile phone prototype*

The video and text for each question or answer are stored on the phone (see Figure 5) and embedded directly in the XHTML pages (see Figure 6 a & b).

JavaScript was then inserted into the XHTML files to control user input, navigation between pages and to store the user input because sending the user input back to a server would require more processing resources and latency.

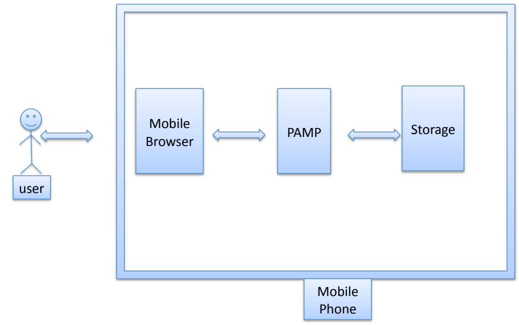

Figure 5: High-level Design of system where the phone is the server and client. The user still uses the mobile phone browser but the content of the system are stored locally on the same phone.

The user input was stored in the browser using cookies. PAMP was used to interpret XHTML source documents to produce the desired web page documents.

C. *Mobile-based prototype interface*

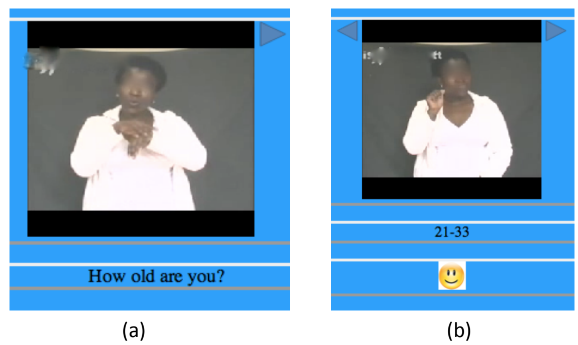

Figure 6: The user interface of the system running in a mobile phone browser. The Deaf user sees the question and clicks on the arrow to the right to get the answer (a); and each answer to the question are shown like in (b).

While capturing the videos these elements were taken into consideration to make the sign language signs more visible: the background had to be darker than the person who is signing, the person who is signing needs to wear one colour, have full control of the lights in the recording room, no accessories to avoid distracting the person viewing the video (see the Figure a).

After the recording was completed, the original video had to be cropped to remove unnecessary things and also to resize the video into a useable video. The video's new height became 660 and its width became 440. The video was then converted into SWF, then embedded into the XHTML file at a square ratio of 190 by 190, as this is suitable for a mobile phone (E71 & N82) screens. Some things cannot be presented with sign language video, the Deaf person needs to remember the medication being prescribed using the other four senses such as seeing, smelling and touching the medication. The doctor's interaction with the system can be seen in Figure 7 (a, b, & c).

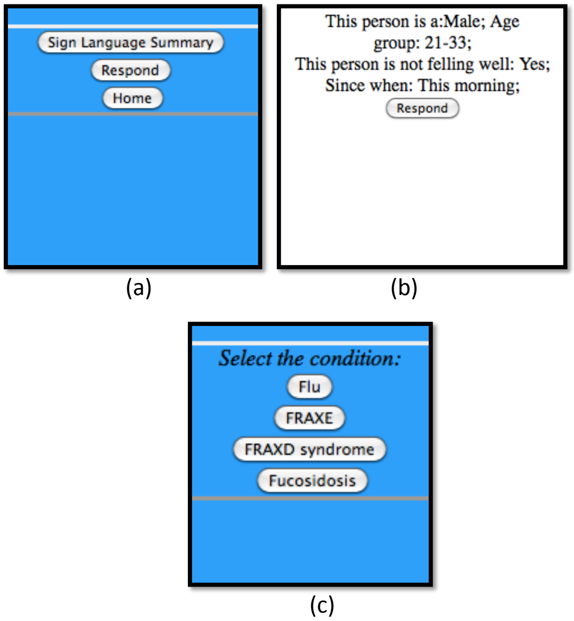

Figure 7: When the Deaf user has completed answering the questions the doctor sees screen (a); screen (b) is the summary of the sign language videos questions; and screen (c) is phrase book with examples of diseases that start with the letter "F".

V. Data Collection and Analysis

A. *Mobile based user testing with the Deaf users*

Table 1: User feedback

Criteria	Deaf user response
Was it easy to use the system?	Definitely Yes
How easy was it to navigate through the system?	I can get used to it, then it will be easy
How easy was it to under what was said in the video?	Very Easy
Would you be happy with video of this quality on a phone?	Very Happy
How clear were the hand gestures in the video?	Very clear
How clear were the facial expressions in the video?	Clear to Very clear
Would you consider using this system in your daily life?	Yes to Definitely Yes
Please specify what you did not like about the system?	1). I get tired of pressing keypads to scroll up and down 2). The screen in small you have to scroll up and down 3). It does not show me a picture of the illness 4). It does not show me a picture of the medication I need to take. 5). The questions take long to answer, too many questions.
Were there any hand gestures that were difficult to understand?	One or Two sign language videos, due to small screen.
How much are you willing to pay for the data charges?	R0,00 to R5,00
Please specify what you liked about the system?	It's a on the phone I am happy with the system.

This table shows the summarised user feedback from both the client server prototype and the client only prototype.

After the mobile-based prototype was completed and loaded on the two Nokia phones (E71 & N82) testing was scheduled with the target users and this was done at the Bastion building. Four members of the Deaf NGO staff tested the prototype. No incentives were given to them as they volunteered to test the prototype. Each person who evaluated the prototype had to sign a consent form. An interpreter was used during the test, to facilitate the communication between the researcher and Deaf users. Since there were four users to test the system, two phones installed with the system were used. The test users were grouped in pairs as they tested the system. Each evaluation began with a storyboard scenario where the system can be used, then the project was explained and then the users tested the system. The feedback obtained from the user trial can be seen summarized in Table 1.

Data capturing techniques used in this user trial were; observation of how they used the system, video recording, questionnaires and discussion at the end of the user trial. Each evaluation with a pair of users lasted for about 20 minutes. The feedback obtained from the questionnaire is summarized in Table 1.

B. *Client-server vs Client only prototypes*

A comparison between the client-server prototype and the client only prototype was carried out and the following results were obtained, see Table 2.

Table 2: Client-Server vs Client only

Criteria	Client-Server prototype	Client only prototype
Platforms	PC, E71 & N82	E71 & N82
Technologies used	PHP, JavaScript, SWF, XHTML, and XML.	XHTML, SWF, JavaScript, and PAMP
Latency	0,3 - 1 minutes to load content on the phone	0 – 0,3 minutes to load content on the phone
Runs in	Mobile browser	Mobile browser
Platforms involved	PC & mobile phone	Mobile phone only
Storage of content	PC	Mobile phone
Network access to content	Wi-Fi	Non
Protocols used to access content	http using Wi-Fi and Apache	http using PAMP
Sign Language video	Understandable	Understandable
Cost for Deaf user	Usual Data transfer costs	R0, 00
Portability	As long as there is a Wi-Fi	Anywhere, anytime

This table shows the comparison between the prototype built to use the computer as a server & the phone as the client; and the prototype that uses the phone only. The system still runs in a mobile browser but the content can between stored on the computer and accessed on the phone or stored and accessed on the phone.

Although both prototypes have the same interface and appear to work the same, they are created and work differently from each other. One of advantages the Client-only prototype has over the Client-server prototype is that it loads its pages faster and this is because it does not require an Internet connection whereas the Client-server does require an Internet connection to load a page, causing a small delay (see Table 2). One of advantages the Client-server prototype has over the Client-only prototype is that, if the user's phone is lost the user can still access the system using the hearing doctor's phone that can handle the system (see Section III.C) whereas the Client-only prototype system is stored locally on the phone thus when the phone is lost the user is left helpless. Both prototypes are accessible to the Deaf user, are equally important and have pros and cons as shown in Table 2.

VI. CONCLUSIONS AND FUTURE WORK

The Deaf to hearing communication aid is a system that allows a Deaf person to construct an English text for a hearing person who cannot sign, by answering sign language video questions presented in an Internet browser. This system is not an expert system (see section II), or related work for more details on expert systems. The system described in this paper is the next step of an evolution of the Deaf communication aid in an Internet browser from the computer-based prototype to the mobile-based prototype. Most Deaf people have capable handsets that they can carry everywhere with them, that they use mainly for SMSs and WAP 2.0. At the current rate in increase of mobile phones among the Deaf, it is possible to deliver the communication aid using Wi-Fi to a Deaf person's mobile phone. A Deaf person having this system on the mobile phone, means that they carry it with them everywhere and can use it in emergencies. This can ease the communication tensions caused by communication breakdowns between a Deaf person and a hearing person. This system can help a Deaf person coupe with an emergency, by getting assistance from hearing person after giving them an English text on the phone describing their situation. This system is not meant to replace the hearing person such as a doctor, pharmacist, train-ticket seller, etc but to help them communicate with a Deaf person. This system is also not meant to replace an interpreter, but to assist in an emergency where an interpreter is not available. The system is a semi-synchronous communication aid, it can be used in a face-to-face communication or as the Deaf person (or hearing person) make their way to a doctor.

This system shows potential to be a useful tool. A continuation of user trials with the Deaf users will follow with the Deaf NGO. Due to the content independent nature of the project, we intend on implementing a Pharmacy prescription on the system, other scenarios can be done as future work. Other aspects of the system that can be considered for future work are the optimizing of video quality, optimizing the video size, and effectiveness of such systems like these among the Deaf communities. In this next user trial we will look at the video quality testing, latency between questions, making the navigation more user friendly and system accuracy. We hope to test the system with both the Deaf person and the hearing person to ascertain for system accuracy.

REFERENCES

[1] Glaser M and Tucker W D (2004). Telecommunications bridging between Deaf and hearing users in South Africa. *In Proc. Conference and Workshop on Assistive Technologies for Vision and Hearing Impairment, (CVHI 2004), Granada, Spain, (CD-ROM publication).*

[2] Looijesteijn K (2009). The design of a Deaf-to-hearing communication aid for South Africa, *Unpublished MSc thesis, Delft University of Technology, Netherlands.*

[3] Krikke J (2004). Streaming video transforms the media industry, *IEEE Computer Graphics and Applications*, 24(4), 6-12.

[4] Segers V and Connan V (2009). RealÂ-Time Gesture Recognition Using Eigenvectors. *Proc. South African Telecommunication Networks and Applications Conference (SATNAC 2009),* Royal Swazi Spa, Ezulwini, Swaziland, 363Â--366

[5] Naidoo N and Connan J (2009). Gesture Recognition Using Feature Vectors. *Proc. South African Telecommunication Networks and Applications Conference (SATNAC 2009)*, Royal Swazi Spa, Ezulwini, Swaziland, 333Â--337

[6] Verlinden SF, and Verlinden HJ (2009). Expert system for medical diagnosis. Google Patents.

[7] John DA, and John RR (2010). A Framework for Medical Diagnosis using Hybrid Reasoning. *Proceedings of the International MultiConference of Engineers and Computer Scientists, 1.*

[8] Duda RO, and Shortliffe EH (1983). Expert systems research. *Science*, 220(4594), 261–268.

[9] AL-Rousan M, Assaleh K, & Tala'a A. (2009). Video-based signer-independent Arabic sign language recognition using hidden Markov models. *Applied Soft Computing*, 9(3), 990–999.

[10] Cox S, Lincoln S, Tryggvason J, Nakisa M, Wells M, Tutt, and Abbott S. (2002). TESSA, a system to aid communication with deaf people. *In Proceedings of the fifth international ACM conference on Assistive technologies* (pp. 205–212).

[11] Power D, Power D M & Rehling B. (2007) German deaf people using text communication: Short message service, TTY, relay services, fax, and e-mail. *American Annals of the Deaf*, 152(3), 291

[12] Zafrulla, Z, Etherton, J, & Starner, T. (2008). TTY phone: direct, equal emergency access for the deaf. *In Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility* (pp. 277-278). ACM New York, NY, USA.

Muyowa Mutemwa holds an Honours degree in computer science from the University of the Western Cape (UWC). He is currently studying for a Masters degree at UWC with the Bridging Applications and Networks Group (BANG).

William D. Tucker is a senior lecturer in computer science at UWC and leads BANG research there. He recently obtained his PhD on the design and evaluation abstractions for information and communication technology for development. One of his field studies concerned technology for Deaf users.