

Improving Usability and Correctness of a Mobile Tool to help a Deaf person with Pharmaceutical Instruction

Michael B. Motlhabi,
William D. Tucker
Mariam B. Parker
Meryl Glaser

Abstract

The computing for development community knows how to build user interfaces using qualitative methods for text illiterate users, especially on mobile devices. However, little work has been done specifically targeting Deaf users in developing regions who cannot access voice or text. This paper describes a multi-disciplinary collaboration towards iterative development of a mobile communication tool to support a Deaf person in understanding usage directions for medication dispensed at a pharmacy. We are improving usability and correctness of the user interface. The tool translates medicine instruction given in English text to Sign Language videos, which are relayed to a Deaf user on a mobile phone. Communication between pharmacists and Deaf patients were studied to extract relevant exchanges between the two users. We incorporated the common elements of these dialogues to represent content in a verifiable manner to ensure that the mobile tool relays the correct information to the Deaf user. Instructions are made available for a Deaf patient in signed language videos on a mobile device. A pharmacy setup was created to conduct trials of the tool with groups of end users, in order to collect usability data with recorded participant observation, questionnaires and focus group discussions. Subsequently, pre-recorded sign language videos, stored on a phone's memory card, were tested for correctness. Results of these two activities are presented and discussed in this paper, in preparation for a future trial in an actual pharmacy.

Categories and Subject Descriptors

K.4.2 [Social Issues]: Assistive technologies for persons with disabilities; D.5.2 [HCI]: User Interfaces—*Evaluation and methodology*; H.5.2 [HCI]: User Interfaces—*User-centered design*

General Terms

ICT4D, Human Factors, Design.

Keywords: Mobile applications, Sign Language, Community-based co- design.

1. Introduction

Mobile devices have found broad application in the provision of medical services and are being used robustly in developing regions [2], not only as communication tools but also as keys to solving socio-economic challenges. For disadvantaged end users,

mobile devices present a significant challenge by way of inadequate literacy skills needed to understand the information on the device [4]. This problem is further compounded for people with disabilities, as can be noted in the context of a Deaf¹ patient visiting a pharmacy to collect medication. Registered sign language interpreters are a possible solution, however they are expensive and thus not accessible to Deaf people because 70% of Deaf people are unemployed thus cannot afford the services of an interpreter [19]. Deaf patients often leave the pharmacy not understanding how to ingest their medication.

Deaf people in South Africa have an average education of Grade 7 [19] and even those who possess rudimentary text literacy experience difficulty in communicating with hearing people. It is a frequent misunderstanding that an Signed Language (SL) is a signed form of a written/spoken language [1]. Thus, most Deaf individuals cannot communicate within the same medium as hearing individuals, and are also unable to read or write adequately. This functional illiteracy renders medicine labels useless to Deaf patients in developing regions, since reading is not a viable option for many of them. Instructions for using medicines are given both verbally, by the pharmacist instructing the patient how to use his/her medicines, and in a written form, by way of a pharmacy-generated medicine label. Since both options are not viable for many Deaf people, a situation may arise whereby a Deaf person does not know how to use his/her prescribed medicines appropriately (see Figure 1). Thus, the need for concise and clear communication of medicine instruction between a pharmacist and a Deaf patient is critical for the process of medication dispensing, especially in developing regions.

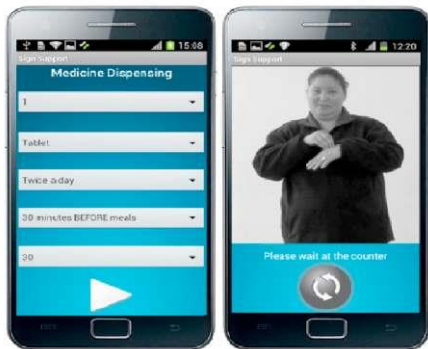


Figure 1: Pharmacist's interface to dispense medication (left) and Deaf person's view of information in SL (right).

This paper is organized as follows. §2 covers background work on which this study is built. §3 discusses the motivation for using information and communication technologies in a developing region, in this case for a disabled subgroup. §4 presents work related to signed language communication aids and mobile videos for signed language. §5 explains the methodology used to realize the current prototype with the

¹Deaf with a capital 'D' is different from deaf or hard of hearing in that Deaf people primarily use a natural sign language to communicate and this defines their sense of culture, similar to other groups who use spoken/written languages.

dual aims to improve usability and correctness. §6 illustrates how SignSupport is used in practice. §7 describes the design and implementation of SignSupport. §8 first recounts data collected from initial role plays to determine the communication options, then analyzes a follow-up trial with Deaf participants and final year pharmacy students and finally, describes the verification testing of the SL videos. §9 concludes and outlines future work.

2. Background

Deaf people in developing regions use services such as Short Messaging Service (SMS) and WhatsApp to communicate with each other and with hearing people. Yet their level of text literacy is adequate for social purposes rather than specific and/or technical discussion [11]. Inadequate text literacy creates a communication barrier, and in a pharmacy environment, only SL can precisely convey information in a way that Deaf people can clearly understand [6]. For these reasons, SignSupport was developed to assist communication between the Deaf community and pharmacists. We collaborate with a non-government organization (NGO), which helps Deaf people deal with social issues. This NGO has approximately 2000 members. All members are fluent in SASL and most of them are functionally text illiterate. Since pharmacists are also the intended users of SignSupport, we also collaborate with local senior pharmacy students as surrogate pharmacists.

SignSupport is a one way communication channel that uses a touch-sensitive interface (see Figure 1) and is based on a system that employed a mock-up of a mobile phone on a desktop computer that allowed a Deaf person and a doctor to communicate with each other using pre-recorded SL videos [15]. The mock-up asked a Deaf person questions in SL. After a Deaf person answered the questions, the answers were presented to a hearing doctor in English. The doctor read the summary of symptoms and responded using an English lookup dictionary. The Deaf person then watched a corresponding SL video.

That mock-up was implemented on a Symbian phone with a guided set of web pages with a combination of SL videos and English text [19]. This enabled a Deaf patient to communicate to a doctor which symptoms they were experiencing and for how long they have been experiencing them. At each step, the Deaf user responded to a series of questions presented in SL, and finally enabled the system to convey how the Deaf user was feeling into English for a doctor to understand and vice-versa. However, the reality of prerecording all communication topics comprising a possible patient-doctor interaction, taking into account all conceivable symptoms and diseases, presented an infinite number of possibilities.

A follow-up effort investigated the way Deaf people use mobile phones to communicate in their daily lives with both hearing and Deaf people. With input from the local Deaf community, SignSupport was re-oriented towards a pharmacy scenario [6]. Since the pharmacy exchange is restricted, pre-recording and storing a

restricted yet useable communication flow on a mobile phone became possible and a team designed an Android prototype [18]. Machine translations of SL are currently problematic. These problems are amplified when there are important structural differences as is the case with spoken/written languages and SL [13]. Automated sign language systems have a limited translation accuracy of merely 61% [10, 20] which is unacceptable for pharmaceutical use. Moreover, using these systems require high-end smart phones or expensive computers [8, 9, 17]. We wish to minimize the cost of a DEV solution by using affordable mid-range technology that will soon become tomorrow's low-end.

3. Motivation

DEV is a still a relatively new field, often driven by Computer Scientists, that explores how computing technologies can be applied to solve the needs of those who reside in disadvantaged regions [4]. DEV has focused on increasing standards of living in developing countries, often by importing or adapting technologies created and used in developed countries. To make the project a success for Deaf people in a developing region, we review a number of factors that could hinder the acceptance of SignSupport: text literacy being the most significant. Mobile phones are heavily text-based. This is a style that was inherited from developed regions, yet many of the people who live in developing regions are text illiterate [16]. Since the target group intended to use SignSupport are not in a position to afford a high-end smart- phone, the phones running SignSupport will be loaned from the NGO. We were conscious of the history of failed ICT4D projects [12]. SignSupport does not use text to communicate with Deaf people but rather SL. Thus, text illiterate Deaf people interact with SignSupport in SL videos and can understand the information being shared with them. This paper demonstrates that it is possible to devise interfaces that can aggregate and accurately represent information for Deaf users in their preferred language, in our case, a signed language.

4. Related work

This section describes the work related to SignSupport and links the principles and methods that others in the area of technologies for Deaf users have used. We study deaf/Deaf technology and how Deaf-specific challenges have been addressed within ICT. Very little work exists for DEV work for Deaf users. Thus, we also review the challenges faced by people with disabilities in developing regions.

4.1 Base technology for Deaf users

MobileASL uses American Sign Language (ASL) and is an ongoing video compression project that seeks to enable low- cost low-bandwidth sign language communication with mobile phone technology [5]. The goal is to make ASL mobile phone communication possible without the need for equipment other than a mobile phone with a front-facing camera. It works on commercial phones that are accessible to Deaf people. The motivation for MobileASL is to make as clear a sign language video as

possible to transmit over a cellular network [5]. Cavendar *et al* conducted user studies with members of a deaf community to determine the intelligibility effects of video compression techniques that exploit the visual nature of sign language. Preliminary studies strongly suggested that even today's best video encoders cannot produce the quality video needed for intelligible ASL in realtime, given the bandwidth and computational constraints of even the best video mobile phones. MobileASL concentrates on three major areas when manipulating video for sign language use; (1) Bitrate, (2) Frame rate and (3) Region of interest. They deemed these three variables important for using sign language videos on mobile phones [5].

4.2 Assistive applications for Deaf users

In the UK, an experimental system called Text and Sign Support Assistant (TESSA) was developed to assist with transactions between a Deaf person and a clerk in a post office by translating the clerk's speech into British Sign Language (BSL) and the Deaf person's signs to text/sound [8]. To generate the signs needed for the TESSA system, the signs of the native signer are first captured as motion data via sensors affixed to the body of the signer. The post office clerk speaks into a microphone, while the system generates the respective signs (currently, not all phrases can be correctly identified) [8, 9, 13], which are represented by virtual signing avatars resembling humans. TESSA was developed for the post office scenario because most of the conversations are predictable and simple to follow. The movements of the avatar are copies of those of a native sign language user. Software specially developed for the project captures the BSL of the signer's hand, mouth and body movements using a variety of electronic sensors. These movements are then stored and used to animate an avatar [8]. The ViSiCAST project sought to improve the quality of life for Deaf people by widening their access to services and facilities enjoyed by the community at large [9]. The objective of the ViSiCAST project was to produce adaptable communication tools allowing sign language communication where only speech and text are presently available. The project identified a number of aspects of life where the integration of Deaf individuals into society would be improved if sign language communication were available, such as access to public services, commercial transactions and entertainment [8, 9]. The ViSi- CAST team started their first prototype with an interpreter and later changed to using an avatar. ViSiCAST project4 now makes sign language communication possible for face- to-face transactions and television broadcasting. One of the aspects of this project transcribes spoken speech to text, which is processed to its equivalent signs in BSL, and consequently signed by an avatar [9]. A face-to-face transaction virtual signing system was tested with TESSA. The biggest drawback of these types of systems is the current lack of sign recognition; their communication is one way, from the avatar to the Deaf person, thus rendering them to be ineffective for critical communication exchange.

4.3 Sign Language translation

A group in Iran uses Artificial Intelligence (AI) techniques to research ways of solving communication problems among Deaf and hard-of-hearing people using mobile

phones. They proposed a system called 'ASRAR', that finds a common factor between Augmented Reality (AR) and Automatic Speech Recognition (ASR) technology which is stored as a string. They combined AR, ASR and Text-to-Speech (TTS) synthesis to develop a system that allows Deaf and hard-of-hearing people to communicate with each other and hearing people [17]. ASRAR gives an opportunity to deaf people to control and manage the information and adapt it easily to a desired form to improve their interactions with hearing people. In the ASRAR system scenario, the ASR engine collects the speech from the detected narrator and uses the ASR engine to recognize the narrator's speech and convert the speech to text. To get the video and the speech of the narrator, the ASRAR uses built-in cameras and microphones on the mobile phone. A joiner algorithm is used to combine AR and ASR engines to work together and does this by updating a version of the text file, which means every word must be saved in the text file. The script operation changes every second when a word is written in the text file by the engine. A survey with Deaf and hearing people was conducted at the end of the research process to understand the interest among Deaf users using different communication methods [17]. Preliminary results showed that the system worked well in many different environments. Deaf participants showed interest in using the system as an assistant to communicate with hearing people.

Like TESSA, SignSupport is designed to function in a limited communication domain setting [18]. For reasons of accuracy, however, SignSupport differs from TESSA in that it does not use Artificial Intelligence. Like MobileASL, Sign-Support emphasizes video quality and resolution, albeit local to the phone and does not yet incur over-the-air mobile data charges. SignSupport uses a similar method as with ASRAR to link English text with a signed language equivalent, although we do not perform any form of automated SL recognition or generation. And like ViSiCAST, we hope one day to generalize SignSupport to handle multiple communication scenarios (see §9).

5. Methods

SignSupport is based on a decade of research and collaboration by a multi-disciplinary team of professionals [6, 15, 19, 18, 21]. §5.1 introduces multi-disciplinary roles. §5.2 explains a design method employed that employs user-centered solutions in a complex design space [3]. §5.3 details the cycles taken to collect requirements and iterate design and development. §5.4 breaks down the video recording protocol and §5.5 details the process taken to verify video content correctness.

5.1 Multi-disciplinarity

Our multi-disciplinary team comprises a broad range of expertise. All team members were involved continuously from design through to development, and from experimentation through to verification.

Deaf participants decide what the project is and how they would like to use it, and most of the user requirements emanate from them because integrating their perspectives increases the chance of developing a successful and accepted system.

Pharmacists play a critical role in the protocol design aspects of the system. Their input brought to bear a number of important pharmacy-oriented directives, concerning:

- Pharmacists' code of practice [7, 21]. Pharmacists were instrumental in designing the application such that it follows a known and standardized logic.
- Ethical principles and respect for persons. As healthcare professionals, pharmacists are trained to adhere to strict ethical standards at all times when interacting with patients. Thus we must ensure correctness of the medication usage and description information displayed in the user interface (see §5.5).
- Data input. SignSupport provides an autonomous means for a Deaf user to acquire medicine instruction, given proper input from a pharmacist. The application had to address the following elements: diseases, dosage forms, medicines, instructions, warnings and recommendations [7].

Industrial design engineers were responsible for determining the appropriate conceptual model of the system. They achieved this goal by involving both Deaf and pharmacy participants, and acquiring requirements by means of role playing, questionnaires and focus groups (see Figure 2). They presented this information in the form of a design/sketch on paper that best represented the expectations of both Deaf users and Pharmacists. They designed Sign- Support's interfaces (Figure 1) based on interactions with end users over several versions starting from the initial mock- up [15], its implementation for medical diagnosis [19] then to the first pharmacy design [6].

A Deaf education specialist was a link between the technical team members and the Deaf community members; a bridge between the technical team and Deaf users of ICT. This specialist helped customize the interface and logic of SignSupport to seamlessly fit Deaf users' expectations, and helped analyze the sentences extracted from the role plays between Deaf patient and pharmacist, and structured the sentences to make sense in SL.

Computer scientists' core duty was to bring the software application to reality, to evolve the human computer interface and verify that the correct SL videos were shown at the right place and time. They examined how end users interface with SignSupport, and helped avoid the problem of overloaded menus by using hierarchical menus [14] as shown on the left side of Figure 1.

5.2 Community-based co-design

The requirements gathering process highlighted four distinct aims. (1) Reliability: since SignSupport is medical software; it is expected to work reliably and accurately without fault at all times (see 5.5 for more details). (2) Usability: both types of users must be able to operate SignSupport with ease after a simple short training session.

There are two different interfaces, one for each user: the pharmacist interface is English-text based and the Deaf interface is SL- video based. (3) Acceptability: users must be willing to use SignSupport. For example, Deaf users need intelligible SL videos and a vibrating *and* an SL-based (as opposed to text) reminder system to help them remember what medicine to take and when to take it. (4) Sustainability: this aim led us to investigate how to easily add more functionality to SignSupport and to design a back-end architecture that will easily accommodate new functionality (see §7 for more detail).

To address these aims in holistic and realistic ways, we employed the strategy of community-based co-design [3]. Naturally, traditional human centered design methods were chosen for the community-based co-design process, as these techniques facilitate the participation of the target groups. This approach required us at every stage to refer back to the participants to show how their suggestions had been incorporated into the SignSupport (see Figure 2). During all interactions with Deaf participants, an SL interpreter facilitated the communication process. Aided by an interpreter, and a co-author acceptably fluent in SL, we could merge the design and usage context by understanding and building positive relationships with the Deaf community. Similarly the communication between the team and senior pharmacy students from a local university were facilitated by a senior pharmacist to build trust and relationships with a local pharmacy community.

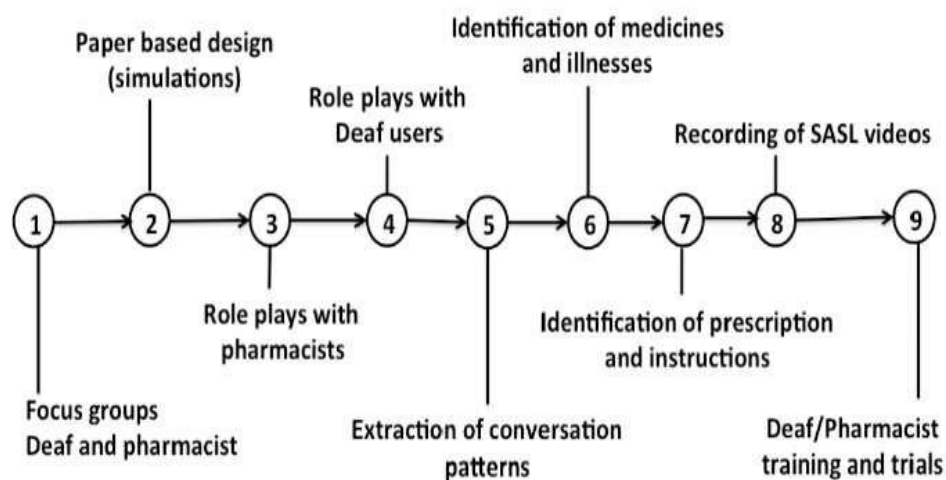


Figure 2: A timeline of events showing how requirements were gathered. Each event with end users also acted as a feedback session showing participants how their prior input was factored into the process.

5.3 Design cycles and development

A cornerstone of the development process was the continuous integration of end-user requirements and feedback into the design process. The result of end-user contributions led to iterative re-design of the entire back-end architecture of SignSupport (see Figures 2 and 8 for more detail). None of the participants were screened by variables including age, gender and race. Instead volunteers were

purposively sampled from the NGO staff because we wanted informed views from the Deaf community. Figure 2 illustrates nine steps which were followed, each described in detail below.

Step 1 comprised focus group interviews with pharmacists. From these discussions, the challenges of dispensing medicines to patients with whom they could not communicate, particularly Deaf patients, emerged because pharmacists use spoken language which Deaf patients are unable to understand. Pharmacists expressed that these interactions were often very difficult, leaving the pharmacist unsure if the Deaf patient had actually understood how to use their medicines. Step 2 was a paper prototype, as a result of Step 1. This prototype exposed basic user expectations for both target users. Steps 3 and 4 comprised role plays to establish user perspectives for both Deaf participants and pharmacists. Since SignSupport is intended for pharmacist-patient interaction, it was necessary to mimic a typical routine interaction between the two users. This was done by studying the patient interaction taught to pharmacy students, based on the School of Pharmacy's Objective Structured Dispensing Examination (OSDE) sheet, a tool used in assessing students on patient counseling. Step 5 was an exercise of conversation mapping. Video footage of the interaction between the two parties during the role plays (Steps 3 and 4) was studied. What was studied was what the pharmacist said to the patient, how it was said, and at which stage of the interaction it was said. We successfully elicited the common dialogues, which occur between pharmacists and hearing patients at public hospitals. The communication flow at the pharmacy was limited in a similar fashion to TESSA, which covered about 90% of the communication at the post office [8]. Step 6 (disease/medicine selection). Pharmacists were asked to help identify one hundred of the most common illnesses they thought were important to include in the prototype. Designations of 47 illnesses were video recorded in SL and stored onto the phone's memory card. Medicine names had to include every possible prescribed medicine for these illnesses. Step 7 (identification of the prescription and instruction). We studied real-world patient prescriptions scripted by doctors and mirrored much of the existing paper prescription layout and content, and optimized it to fit on a mobile device. The prescription text, instructions on the prescription and the sequence in which these instructions occur were reviewed and incorporated into SignSupport. This was done to ensure that when the pharmacist dispenses, s/he follows an already familiar natural flow. Note that while SignSupport can act as a virtual prescription, it is not intended to replace the doctor or pharmacist. Step 8 (SL video recording): A finite set of 180 videos were recorded to represent the possibilities of diseases, medicines and instructions determined in Steps 6 and 7 (see Figure 8). A conversation script was created and used to guide the recording of the SL videos. An interpreter translated each message together with an informed Deaf staff member of the Deaf NGO. Step 9 (training and trial): Deaf and pharmacist participants underwent training, in two separate groups of 8, on how to use SignSupport. Each session was about three hours and included a projected presentation followed by hands-on practice with SignSupport running on 8 phones. Participants were encouraged to "play" with the

application and provide feedback after their hands-on usage. Each training session was video recorded. Subsequently, both groups then participated together in a test trial at a mock hospital pharmacy. Pharmacists had to dispense medicines as per actual prescriptions to Deaf patients the simulated dispensary. Care was taken to mimic, as far as possible, the scenario that would occur when a Deaf patient collects medicines at a hospital pharmacy, including not being able to hear their name being called when it was their turn. Deaf patients were asked to present their prescriptions to a pharmacist at the counter and s/he used SignSupport to dispense the medication. After participants finished the trial, they were asked to complete a questionnaire individually, and then participated in a combined focus group discussion where they could give more detailed and open-ended feedback.

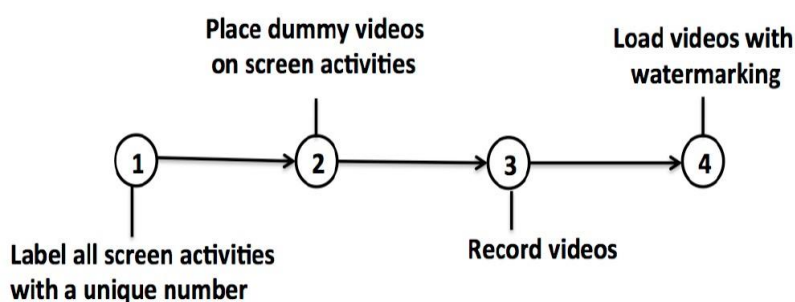


Figure 3: Steps showing how the initial SL video recordings transpired.

5.4 Initial verification during recording

SignSupport is heavily dependent on the SL videos that are pre-loaded onto a phone's memory card. It is therefore imperative that the correct video corresponds exactly to the English text instructions. The first level of verification occurred when initially recording the videos. A follow up verification procedure is described in §5.5. We devised a set of rules/procedures as seen in Figure 3 to verify that the videos actually represent the instructions that the pharmacist has communicated. The following steps were taken to record, edit and load the videos onto SignSupport, as illustrated in the figure.

Step 1: Number every screen activity on SignSupport that will display a video. Every number corresponds to a sentence that needs to be recorded on the content script (a script containing all the sentences that need to be recorded in SL).

Step 2: Label every screen activity on SignSupport with a dummy/placeholder video that explains in English text what SL message will be displayed on that screen.

Step 3: Once all the activities are labeled, numbered and appear in the correct place, start recording videos, reading from the content script and translating the text to SL with the help of a SL interpreter. Every video that is recorded is watermarked with the corresponding number from Step 2.

Step 4: Edit the videos, label them and remove the placeholders within SignSupport, then load the actual (not dummy) videos onto SignSupport one at a time using the watermarking numbers from the previous step. There are two ways in which we test whether the correct video has been displayed; this is explained in the next section.

5.5 Subsequent verification procedure

This section details the methods used to verify that Sign- Support communicates the correct sign language information for medication that the pharmacist has conveyed in text. Figure 4 below shows the sequence of taken to verify the correctness of videos within SignSupport.

Below is a breakdown of Figure 4. We focus on two parameters.

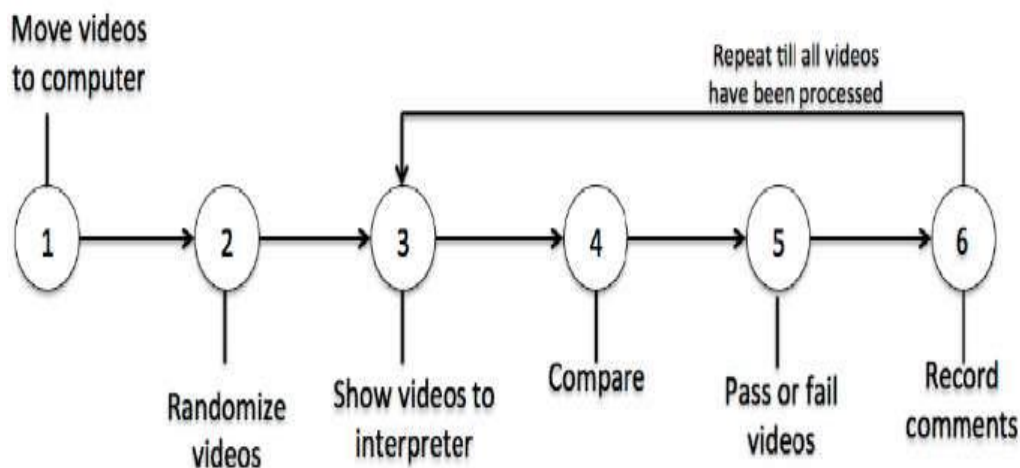


Figure 4: A diagram showing steps taken to verify correctness of the videos.

Parameter 1 tests for the content of the video, and parameter 2 checks if the video appears in the correct place. We know what the content of the video should be from the conversation script and we know the position of the video from the watermarking.

Parameter 1 follows these steps:

Step 1: Transfer all videos from the phone's memory card onto a computer's hard-drive.

Step 2: Randomize the videos (arrange them in a way that it is not possible for the interpreter to predict the next video) and give them unique numerical identities.

Step 3: Present the videos to an interpreter who watches them on a different computer monitor on the other side of the desk and voices their content in English. Here we are expecting a general translation of what the video explains.

Step 4: After the interpreter watches the video and an explanation is given, we look at the conversation script that has the two variables on it. Variable one is the number identity of the video. Variable two is the English equivalent of the SL used in the video. So to mark a video as correct/satisfactory, we look at the number given to that video, listen to the interpreter's explanation and compare it with the one on the script.

Step 5: If a match is found in Step 4 we tick the "pass box"; otherwise the "fail box" if no match is found.

Step 6: Record comments from the interpreter on how some of the content could have been expressed differently in SL for the next system iteration (see §/refconclusion). Repeat Steps 3 through to Step 6 until we have viewed and checked off all the videos that are on the system.

Parameter 2 is a simulation exercise and involves using SignSupport to verify the position of the videos, and follows the following steps:

Step 1: Conduct role plays with the interpreter by running through the entire application. At each stage where we encounter an SL video, s/he interprets it and we confirm with the script that indeed the video means what it should and appears where it should. We perform this task in the same way a Deaf user using SignSupport would.

Step 2: Test the medicine instructions from the pharmacist. We prepared "dummy prescriptions" that covered all of the different instructions and permutations that the pharmacist can set. From this comprehensive list we randomly chose the prescriptions that would be entered in SignSupport. After we entered those, we asked the interpreter to explain the message on the SL videos. Here we were testing whether the English text selections made by the pharmacist are consistent with the information on the videos, referring to the conversation script for confirmation.

6. Using the signsupport in practice

This section summarizes how to use our system. SignSupport requires an Android phone running at least version 2.3.3. Once installed, SignSupport opens by finger-tap. Figure 1 shows two typical interface screens, one for the pharmacist and one for a Deaf user. Any Deaf user associated with the NGO could conceivably borrow the phone when going to the hospital pharmacy and return it when medical treatment is completed. What follows is a typical scenario as described more fully by [18].

At the hospital, a doctor diagnoses the Deaf patient and hands him/her a paper prescription as per normal. The Deaf patient takes this prescription and the smartphone with SignSupport to the pharmacy. While waiting for the prescription to be processed at the pharmacy, the Deaf user can enter background information including medicine allergies, gender, access to clean water, pregnancy status and other information that the pharmacist may require to dispense medication

appropriately. The background information complements the information about the patient on the patient card at the pharmacy.

A Deaf patient must unlock SignSupport by entering a four-digit PIN that protects the patient's medical information. When the Deaf patient is called upon to retrieve his/her medicines, usually by hand because Deaf patients are usually shown to a special needs queue, s/he hands the phone to the pharmacist as soon as s/he gets to the dispensing counter. At this point, the interaction between the pharmacist and Deaf patient commences using SignSupport as the communication medium. The first screen prompts the Deaf patient to show a hospital identification card. The pharmacist is then able to ensure it is the correct patient, and can view the patient's background history and can check for problems like allergies and concomitant medical conditions before dispensing the medicine. The pharmacist interacts with the application's interface in order to dispense the prescribed medication, as shown on the left side of Figure 1, by tapping information on the phone's display, selecting from the provided options and capturing a photograph of each medicine with the phone's built-in camera.

Since every screen activity is in line with the accepted pharmacy practice code [21], this process should be easy and natural to follow for a trained pharmacist. Information about each prescription is delivered in SL videos for the Deaf patient. SignSupport allows the Deaf patient to review instructions for any prescribed medication in SL at any time. Currently, SignSupport also reminds the patient when to take their medicine in text (see §9 below), and also warns users when they are about to run out of medicine for chronic conditions such as hypertension, diabetes or cancer. Note that the text phrase beneath the video is not meant for the Deaf user, and is a key phrase to help a non-signer follow the application logic.

7. Design and implementation

This section breaks down technical details of SignSupport, including its evolution from the first prototype to the current more robust and flexible user interfaces. The version of SignSupport presented in this paper has undergone three iterative cycles of development. Each stage of development brought changes to the SignSupport's back-end navigation

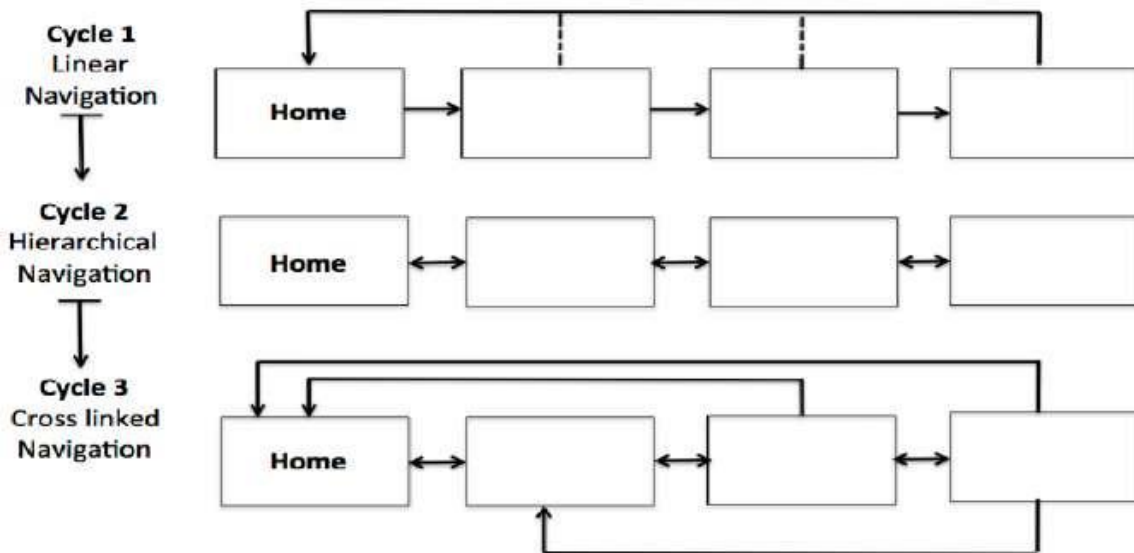


Figure 5: A representation of the user interface navigation used for SignSupport.

structure (see Figure 8) which in turn changed the flow of the user interface (see Figure 5). The block structures in Figures 5 and 6 represent screen activities within tSignSupport and the arrows represent the direction of flow for all screen activities.

7.1 System back-end evolution

In Figure 5, the downward pointing arrows demonstrate how the system has developed over successive generations, becoming more robust from Cycle 1 through to Cycle 3, the current development cycle.

Cycle 1: After users had been interviewed via focus groups and had drawn their own depiction of how the solution should look, our resident design engineer sketched a design that was coded and deployed on an actual device. The first prototype was monolithic and used a linear approach to system navigation(see Figure 5). This approach was acceptable for small applications with about three or four screen activities. However, it was undesirable eventually deemed unacceptable as the current prototype contains over fifty screen activities. Moreover, people made errors and needed to go back to a previous screen. The linear navigation approach gave us an idea of how to organize and structure screen activities, so that we could start to address them in a more efficient way.

Cycle 2: Hierarchical navigation was introduced in the second iteration (see Figure 5), and came in two forms: when a screen activity had multiple drop-down menus SignSupport as can be seen in Figure 1, and between screen activities that appear to the user at different stages. Although there were two different implementations for hierarchical navigation, the application algorithm was the same for both. The difference was how we aggregated the data that had been entered into the system. The case of moving from one video-playing activity to the previous or next proved to be

challenging because of the high frame rate and resolution of the SL videos on the memory card. Limited processing power of the mobile device caused the application to intermittently fail when loading the videos. We later discovered that the *AndroidMediaPlayer* programming interface caused the error, as it could not play videos back-to-back on different screen activities. We remedied this problem by creating a buffer activity that separated the video activities and allowed the device to redistribute its resources/memory and prepare it for the next video activity. When collecting data from multiple screen activities, the data from the user was written to a file (similar to ASRAR) and kept there until all of the instructions/tokens were collected (Figure 7), and at each stage the user could go to the previous activity and re-enter an instruction (see Figure 5). Should the user make an error, it was not necessary to go to the end and repeat the entire process, and s/he could go back one screen activity at a time. The former design would not be suitable, as it wastes time and led to user frustration.

Cycle 3: Cross-linked navigation is currently a combination of the two approaches discussed above. The system has areas where a monolithic navigation style is required. This is mostly true for the Deaf user interface, e.g. the one-time only set-up of patient background information. Hierarchical navigation was used for pharmacists because of the playback challenges mentioned in Cycle 2. The system's back button could not be used for this cycle because it does not call the buffer activity but "calls" the *AndroidMediaPlayer*, which is not suited to our needs. To solve this problem, we created a softkey button inside the application and disabled the device's "Back" and "Menu" buttons. This means navigating the system is only possible from inside the application. We can control not only the direction of flow but also the process that leads to a specific event. The cross-linked approach also increases the productivity of the pharmacist during dispensing as it allows the user flexibility to move back and forth, and make adjustments spontaneously. Thus far, we have explored a number of different designs as seen in Figure 5 with different outcomes. The current design is therefore a culmination of three cycles of both technical and user-centered design and evaluation.

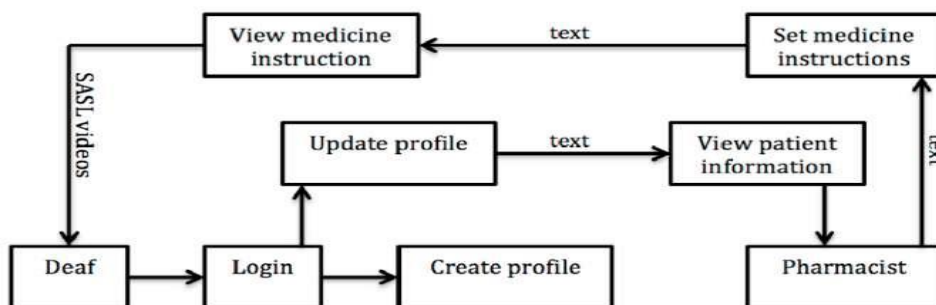


Figure 6: The Deaf user provides text to the pharmacist via SL interaction and the pharmacist provides SL to the Deaf user via a graphical user interface with text

Figure 6 shows a high-level use case of the entire system and its users. From Figure 6 we can see that the system forms a closed loop. This is why we define the system as a limited communication device tool. Communication is deliberately limited to the pharmacy context and is for the most part pre-recorded on the mobile device running system.

7.2 System specifications

Effort was dedicated to the provision of acceptable video quality to SL users. We edited the video size to be 640x700 (width and length) in MPEG-4 format at 30fps. We chose 640x700 dimensions because we can programmatically manipulate the video length and width (dimensions) without distorting the signed language video when viewed on differently sized screens. Thus, the bigger the phone's display the better the sign language video quality. MobileASL found that at between 10fps and 15fps users could not distinguish the difference in video quality [5]. While this is true, the video quality can still be poor at such low rates. A higher frame rate helps with legibility [10, 5] during video compression from MOV to H.246. We convert from MOV because that is the format that our recording camera uses to H.246/MPEG-4 AVC (Advanced Video Coding), which is a standard of video compression that uses open source and also because MPEG-4 is the official video format for Android OS. We have coded the system to fit on most currently available Android phones.

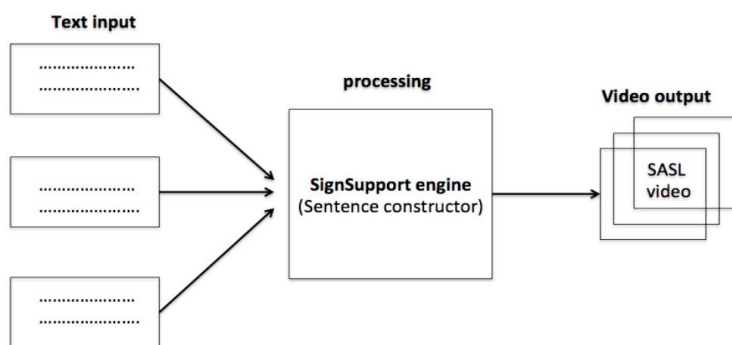


Figure 7: A diagram showing the Sentence constructor on system

The mobile phones we used had at least 2 gigabytes (GB) of external storage space or more, 1GB of RAM and a back-facing camera. The SL videos are an average length of about 2:00 minutes and they were all recorded at the same location with the same background and lighting conditions over two days. We removed the sound from the videos to make the videos smaller and set all videos to black and white as suggested by Looijesteijn [15]. This is because pixels have only one property, i.e. colour. The colour of a pixel is represented by a fixed number of bits. The more bits, the more subtle variations of colours can be reproduced and thus the larger the video. For this reason we chose to make the videos black and white. This allowed us to essentially give each pixel either a black or white colour resulting in a smaller video.

The user interface was coded with Extensible Markup Language (XML), a language that allows the definition of tags, while having the qualities of HTML [www.w3.org/XML].

7.3 Text-to-SL

The pharmacy backend is coded in Java and has four layers as seen in Figure 8.

Layer 1 contains possible medical conditions, currently restricted in number to code and trial the prototype. Layer 2 holds the medicines in the system, similarly restricted. Videos from Layers 1 and 2 appear on different screen activities at different times, so these parameters are labeled with the same name with which the SL videos are stored. To fetch and play them, we only reference the name of the disease or medicine to the correct directory on the memory card. Layer 3 holds combinations of prescription instructions with different permutations. Videos in this layer are recorded as complete sentences in SL. A selection of one item on every axis forms a token is. This token written to a file that is later accessed and read (similar to ASRAR). We aggregate the data contained in these files to form a token sentence. This sentence matches one of the SL videos on the memory card. We search the memory card using linear search because the videos are fetched randomly depending on the prescribed medicine. We find a particular video by performing linear search on the memory card and comparing every string token until we find the one that matches. The matching string is the name of the video and that is the video to playback for the Deaf user. For videos that appear in a predictable sequence (background and security videos) we play them back by calling their associated text names (no searching/sorting is necessary here). However this is impossible to achieve with the rest of the videos on the system because we do not know beforehand what video will be played next. This is why a video order procedure is needed and implemented. Layer 4 holds combinations of possible warnings and recommendations for the Deaf end user and uses the same simple lookup algorithm as for Layers 1 and 2.

A Deaf patient reviews a prescription in a four-stage sequence. For videos in Layer 3, we recorded a limited number of complete sentences instead of stitching together fragments because they would not make sense in SL. We concentrated on three prescription factors: frequency, quantity and dosage event. Selected values for the three parameters limit the pharmacist from making selections that are not pre-loaded and thus limiting the communication flow. We can restrict the domain of communication because we have captured all the conversations and their flow loaded on the phone [18]. The TESSA research group studied restricted domain communication and covered about 90% of the exchange [8] that occurs at a Post Office, we have used some techniques deployed in the TESSA project to also ensure that we reach high percentage levels of the communication exchange in the pharmacy context.

We store all SL videos on the phone's memory card so as not to incur network charges. This means all communication is effectively limited to what has been stored on the phone, and does not cost the Deaf user anything at all to use the application.

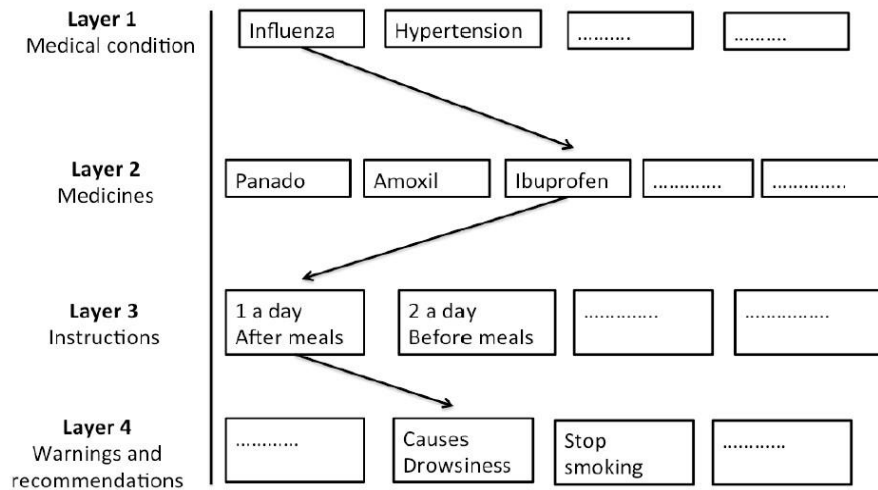


Figure 8: A layered view of the pharmacy backend that indicates how a prescription is encoded.

8. Results and analysis

This section provides and analyses the results obtained from the most recent development cycle consisting of two types of testing indicated above: usability of the user interface for both types of users, Deaf and pharmacist, and the verification of the videos; that they say what they are supposed to say, and do so correctly according to the application flow.

8.1 Usability

Participants (n=16) were present for training and testing: semi-computer literate Deaf participants (n=8) from a local NGO and senior pharmacy students (n=8). The sessions were conducted using role plays followed by individual questionnaires and then focus group discussions. We used SL interpreters to collect data from the Deaf participants. The focus was on usability testing including monitoring user interaction with the system, and identifying potential design flaws to be addressed in the next prototype. Researchers did not assist any of the participants during the role plays. Deaf participants were asked to input background information into the system while they waited for a prescription to be handed to them. Pharmacists worked at the dispensary counter as they normally do, and patients were motioned to the front to collect their medicine. When they got to the counter, they produced the prescription and phone, and handed both to the pharmacist. The pharmacist (student) used the phone and the (faked) doctor's prescription to dispense medication, without directly communicating with the Deaf patient because that was impossible as none of the pharmacists knew SL.

Both sets of participants were asked to answer a questionnaire that enquired about the usability of the software and also what they would like to improve. Pharmacists reported that the system was easy to use. They suggested that it was much better to use the SignSupport to dispense medicine to a Deaf patient. The average dispensing time using Sign- Support was 4:23 minutes. In the first run of role plays, pharmacists dispensed medicine without SignSupport and their average dispensing time was about 9:55 minutes per patient. Pharmacists reported that SignSupport was direct and to the point about giving explanations and instructions.

Deaf users were happy to use SignSupport for collecting medicine. They reported that it was easy to use and they would use it in real life. Deaf participants did not have any challenges navigating, performing and completing tasks using SignSupport, and this they did after just one training session. All the Deaf participants accepted SignSupport, but they expressed concern that pharmacists would not accept the software at real pharmacies. We explained that the hospital staff would be informed of the technology when it is ready for deployment.

8.2 Video verification

To establish SL video correctness, we were not testing SignSupport itself, but rather using it as a tool to verify the content and position of the SL videos within the prototype. This section details the results following the procedure outlined above to test whether the videos on SignSupport give the correct information about the prescribed medication. The following tools were used; a computer, a monitor, a video, an interpreter, a conversation script and an Android phone running SignSupport. The system was divided into two: background setup and pharmacy dispensing. Below is an analysis based on evaluating both of them.

The background setup contains 18 sign language videos. These are short sign language videos used to extract personal information from a Deaf user when using SignSupport for the first time. All videos in this section passed the verification procedure and could be understood by the SL interpreter, and confirmed with the corresponding information on the conversation script. All videos found to be in the correct position within the prototype.

For pharmacy-dispensing, there are 162 instruction videos and 35 videos were found to be either undecipherable, ambiguous or the semantics did not match the conversation script. Most of them could be understood at first glance, some were unclear because of the signs that were used and others were discovered to be unusable because they did not convey the information in the most understandable/desirable format. However, they were all found to be in the correct position within the prototype. All participants showed a positive response during the training and testing process and were always eager to try out the exercises with which they were presented. The learning curve of the participants was remarkable since none of the participants raised any major questions about the functionality of the

system. The video verification test showed that out of 180 videos only 35 could not be used and require re-recording. This translates to 80.6% usable videos. We found that all videos that deal with a patient consuming one tablet/capsule a day were misleading and thus could not be used consistently. For example, the video saying "take one tablet once a day every 24 hours after meals" was interpreted in SL videos as "take one tablet every morning once a day every 24 hours, every 24 hours". Furthermore, "take one tablet every 6 hours four times a day after meals" was interpreted as "take a tablet four times a day after meals, 6 hours after, 6 hours after, 6 hours after, 6 hours after (sic)". One interpretation is that the repetition is for emphasis. However, the senior pharmacist insists that it could lead to patients overdosing.

9. Conclusion and future work

SignSupport has shown promise as an effective communication bridge between a hearing pharmacist and a Deaf patient. The following elements of the application can be enhanced to ensure its success in a real-world situation. All SL videos must be verified to be a direct 100% translation of the English instruction. Furthermore, the positioning of the videos must be re-established after re-recording in order to continually ensure the correct dispensing sequence. Malfunctions in either of these components can result in harm to a patient. Since 19.4% of the recorded videos were found to be either ambiguous or erroneous, these videos will be re-recorded with the presence of a pharmacist, an SL interpreter and an informed Deaf member from the Deaf NGO. A subsequent verified version of SignSupport with corrected videos will be assessed by experimentation in an actual hospital pharmacy after sufficient medical ethics clearance has been granted.

SignSupport is an "internal sign language translation" system, internal in the sense that it incorporates a closed loop of limited conversations which typically take place between pharmacist and patient. SignSupport does however not include some conversations that could possibly arise, for example a Deaf patient asking a question. SignSupport is designed in such a way that it answers most common questions asked by patients before they actually ask the questions, for example, recommendations and warning explanations, when to come back for a refill and so on are some of the common question which SignSupport can answer. In the future to solve this problem SignSupport might include functionality that allows for a video relay break-out, which will allow a video conference-type communication, whereby an off-site interpreter can relay SL to English, for example, and vice versa.

Although SignSupport was designed, implemented and tested in a pharmacy context, it could also be modified and applied to any context, e.g. a Police Station or Home Affairs. Furthermore, we also realize that this tool could be adapted to use any signed language, and could also be used with audio instead of video to serve the text illiterate in developing regions everywhere. These generalizations would entail building an authoring tool to allow SignSupport to be context and (signed) language independent,

and would make an even more valuable contribution to the computing for development community.

10. Acknowledgments

We thank the Deaf Community of Cape Town for their collaboration, and SANPAD for financial support. Thanks also to Prangnat Chininthorn, Adinda Freudenthal and to all interpreters who have helped us with this project. We also thank Telkom, Cisco, Aria Technologies and the THRIP (Technology and Human Resources for Industry Partnership) initiative of the South African Department of Trade and Industry for financial support via the Telkom Centre of Excellence (CoE) programme. The National Research Foundation (NRF) manages THRIP funding. Any opinion, findings and conclusions or recommendations expressed in this material are those of the authors and therefore the NRF does not accept any liability in regard thereto.

11. References

- [1] D. Aarons and M. Glaser. A Deaf adult literacy collective. *Stellenbosch Papers in Linguistics*, 34:1-18, 2002.
- [2] Y. Anokwa, N. Ribeka, T. Parikh, G. Borriello, and M. Were. Design of a phone-based clinical decision support system for resource-limited settings. *In Proc. ICTD*, 2012.
- [3] E. Blake, W. Tucker, M. Glaser, and A. Freudenthal. Deaf telephony: Community-based co-design. In *Interaction Design: Beyond Human-Computer Interaction*, pages 412-413. Wiley, 2011.
- [4] E. Brewer, M. Demmer, B. Du, M. Ho, M. Kam, S. Nedeveschi, J. Pal, R. Patra, S. Surana, and K. Fall. The case for technology in developing regions. *Computer*, 38(6):25-38, 2005.
- [5] A. Cavender, R. Ladner, and E. Riskin. MobileASL: Intelligibility of sign language video as constrained by mobile phone technology. *In Proc. SIGACCESS*, page 78, 2006.
- [6] P. Chininthorn, W. Tucker, A. Freudenthal, and M. Glaser. Mobile communication tools for a South African Deaf Patient in a Pharmacy Context. *In Proc. IST-Africa*, 2012.
- [7] S. A. P. Council. *Good Pharmacy Practice in South Africa*. Printed By S.A Council, 2010.
- [8] S. Cox, M. Nakisa, and M. Lincoln. TESSA, a system to aid communication with Deaf people. *In Proc. SIGACCESS*, pages 205-212, 2002.
- [9] R. Elliott, J. Glauert, J. Kennaway, and I. Marshall. The development of language processing support for the ViSiCAST project. *In Proc. SIGACCESS*, pages 101-108, 2000.
- [10] M. Ghaziasgar and J. Connan. Investigating the intelligibility of synthetic sign language visualization methods on mobile phones. *In Proc. SAICSIT*, pages 86-92, 2010.
- [11] M. Glaser and W. Tucker. Telecommunications bridging between Deaf and hearing users in South Africa. *In Proc. CVHI*, 2004.
- [12] R. Heeks. Information systems and developing countries: Failure, success, and local improvisations. *The Information Society*, pages 101-112, 2002.
- [13] M. Hilzensauer. *Information Technology for Deaf People, Center for Sign Language and Deaf Communication*. University of Klagenfurt, 2006.
- [14] M. Jones and G. Marsden. *Mobile Interaction Design*. Wiley, 2006.
- [15] K. Looijesteijn. *The design of a Deaf-to-hearing communication aid for South Africa*. Unpublished MSc thesis, Delft University of Technology, Netherlands, 2009.
- [16] I. Medhi and K. Toyama. Full-context videos for first-time, non-literate PC users. *In Proc. ICTD*, pages 140-148, 2007.
- [17] M. Mirzaei, S. Ghorshi, and M. Mortazavi. Helping Deaf and hard-of-hearing people by combining augmented reality and speech technologies. *In Proc. 9th Intl Conf. Disability, Virtual Reality & Associated Technologies*, pages 149-158, 2012.

- [18] M. Motlhabi, M. Glaser, M. Parker, and W. Tucker. SignSupport: A limited communication domain mobile aid for a Deaf patient at the pharmacy. *In Proc. SATNAC*, pages 173-178, 2013.
- [19] M. Mutemwa and W. Tucker. A mobile Deaf-to-hearing communication aid for medical diagnosis. *In Proc. SATNAC*, pages 379-384, 2010.
- [20] T. Starner and A. Pentland. Real-time American Sign Language recognition from video using HMMs. in motion based recognition. *In Proc. 2nd Intl. Conf. Computer Vision*, pages 222-243, 1997.
- [21] J. Watermeyer and C. Penn. Working across language and culture barriers: Communication skills for pharmacists. Technical report, Johannesburg, South African, Printed in Cape Town 2009.