DEAF TELEPHONY: COMMUNITY-BASED CO-DESIGN

Edwin Blake, William Tucker, Meryl Glaser and Adinda Freudenthal

1. Context

The Deaf Telephony project set out to assist South African Deaf people to communicate with each other, with hearing people and with public services. The team currently comprises researchers from the University of Cape Town (UCT), the University of the Western Cape (UWC), the Technical University of Delft (TU Delft) and the Deaf Community of Cape Town (DCCT—an NGO). This team has been working for many years with a Deaf community that has been disadvantaged due to both poverty and deafness. The story of this wide-ranging design has been one of continual fertile (and on occasion frustrating) co-design with this community. The team's long-term involvement has meant they have transformed aspects of the community and that they have themselves been changed in what they view as important, and in how they approach design.

Deaf users in this community started out knowing essentially nothing about computers. Their preferred language is South African Sign Language (SASL) and this use of SASL is a proud sign of their identity as a people. Many are also illiterate or semi-literate. There are a large number of Deaf people using SASL; in fact there are more than some of the smaller official languages. Since the advent of democracy in South Africa in 1994 there has been an increasing empowerment of Deaf people. SASL is accepted as a distinct language in its own right. Deaf people encounter many problems with communication. Currently communications technology does not support sign language.

2. Method

In this case study we present an overview of the action research and design cycles over a period of ten years (2002–2011) during which both technological solutions and methodology were developed. Together with the Deaf community we have, over the years, evolved a way to collaborate for mutual benefit. Action research gradually moved from prototypes for research towards interventions that are used daily at the Deaf community centre. Our case study shares these learning experiences about the changing action research methodology.

The eventual method shares aspects of community-based co-design and action research. This methodology is a way of exploring a design space in a way that alleviates the restrictions of the designer's own viewpoint and bias. In a cyclical fashion the designers develop according to their skills and learning and according to the users' expressed requirements and their learning. The researchers and the users end up being the design team.

3. The Development of Community-based Co-design

The current combined approach, fusing action research, industrial design approaches, education and other societal measures was named 'Community-based Co-design'. An initial validation of its feasibility was a joint review meeting of research and members of the Deaf community held in 2010. We evaluated the list of projects presented below and found directions for next steps.

The intention of the meeting was to launch the next phase as equal partners. It was clear that for the Deaf community the concerns were frequently of a more immediate nature (for example, an expansion of training in literacy and ICT, which is now emphasized in the forthcoming phase). The researchers would still lead action research measures which aimed at technology delivery in a more distant time but directions would be shaping jointly.

A discussion about how to approach and influence government policy was debated. A tentative outcome was that the most effective method of influencing policy would be to empower Deaf people to communicate their own requirements in interactions with government. Providing communication tools would be a significant aspect of empowerment.

4. Future Directions

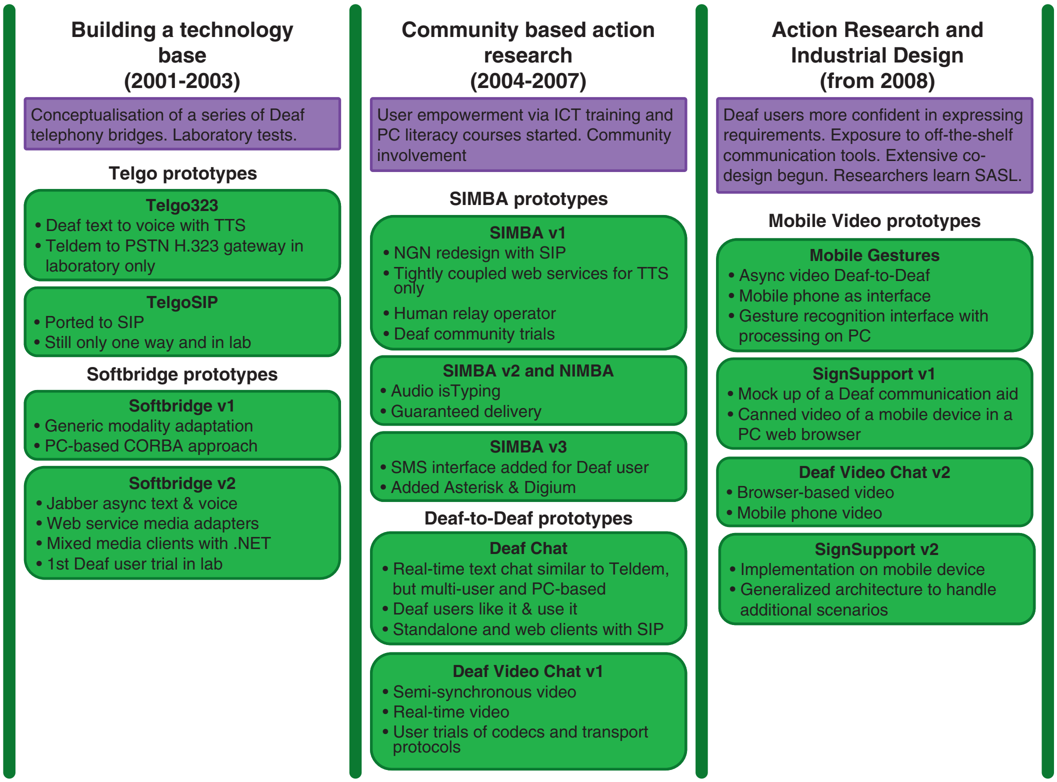
The joint review meeting mentioned above set our agenda for the coming three years, in terms of approach, of roles and in specifying directions.

Our work is driven by the need for Universal Access for communication. From our earlier experiences we now believe that besides practical (technological) solutions and education, it is also imperative is to drive policy implementation. The South African framework of legislation comprises excellent policy on access. South Africa respects the right to be helped by an interpreter, and technology should ensure inclusive use of public amenities by all consumers. However, enacting policy is also about implementing practical solutions. In practice expensive and scarce human interpreters, which are currently the only option, are often not available. Technology (plus education) can be much cheaper than human labour, thereby providing promising solutions. Cost will remain an obstacle even at the more affordable rate offered by new technology. Deaf communication would be greatly aided by lower rates for mobile video streaming—and new government regulations can facilitate this.

Therefore, in the currently ongoing community-based co-design work we do not only target technology research and design, but we address three topics at the same time:

- Technical development
- Influencing government policy concerning Universal Access to communication
- Capacity Building—ICT training

This long-term intimate involvement with the community has moved us from an ethical position where the researchers are the 'experts' and 'active agents' obliged to deal ethically with 'subjects' or 'users' to one where the major ethical consideration is one of reciprocity between equal partners. A long collaboration and mutual trust is needed to learn to collaborate like this.



An overview of the action research cycles. Click on the cycle to view more detail

5. The Case Study

The diagram highlights the various stages of our research project and lists the various prototypes that formed nodes in a design trajectory. The cyclical methodology of honing in on an effective implementation can be seen.

The Deaf telephony prototypes are grouped according to our basic methodological stance and then along several architectural themes.

The initial project aims were informed by outcomes from Teldem trials conducted by Glaser with a local Deaf community (Glaser, 2000). The methodology here was the user-centred design within an agile software engineering approach. The Telgo prototypes bridged a Teldem text terminal to a voice device on the Public Switched Telephone Network

(PSTN) with Text-To-Speech (TTS). The Softbridge prototypes provided fully automated Deaf-to-hearing bridging between text and speech using Instant Messaging.

Based on user interaction we moved towards a community-based approach where we met the community demands for greater empowerment with regard to technology. At the same time the system testing was done within the community at their centre. The SIMBA prototypes continued the theme of using Instant Messaging but opted for a human relay operator rather than Automatic Speech Recognition (ASR). The text chat prototypes provided synchronous text messaging, similar to the Teldem, but with Internet clients. The video chat prototypes explored asynchronous video messaging to support high quality sign language communication.

Finally we adopted a co-design methodology within the community from about 2008. This resulted in greater exploration of the user lives and lifestyles. Researchers and students all learnt South African Sign Language as part of their involvement with the community.

5.1 Building a technology base (2001-2003)

Telgo323
- Deaf text to voice with TTS
- Teldem to PSTN H.323 gateway in laboratory only

Telgo323 Telgo323 was our first attempt at a Deaf telephony solution. During this cycle, we engaged an intermediary to the Deaf community rather than the community itself. Together we devised an overview of telephony options for members of DCCT (Glaser & Tucker, 2001). An MSc student at Rhodes University designed and built an initial prototype (Penton et al., 2002). Telgo323 converted a Deaf user's text on the Teldem to speech on the telephone. We felt that a two-way conversation was necessary to demonstrate the prototype to Deaf users, and therefore did not show it to them. The following table provides an overview of the cycle.

Cycle overview	Description
Timeframe	Early 2002
Community	N/A
Local champion	N/A
Intermediary	Meryl Glaser (UCT)
Prototype	Telgo323
Coded by	Jason Penton (Rhodes)
Supervised by	William Tucker (UCT/UWC), Alfredo Terzoli (Rhodes)
Technical details	(Glaser & Tucker, 2001; Penton et al., 2002; Penton, 2003; Tucker et al., 2002)

Telgo323 cycle overview

Diagnosis

A prior Teldem field study showed that ICT could be designed with adequate functionality and the best of intentions yet still not be adopted by the target community (Glaser, 2000). Deaf users approved of the Teldem concept, but could not, or would not integrate the Teldem into their lives. The Teldem exhibited frequent technical faults that required a tedious reset process. It was difficult to discern if the displayed text came from a person or the Teldem itself. If the former were true, there was no way for a Teldem user to identify the other caller. When the latter occurred, the text was often incomprehensible. There were also financial issues with the Teldem. A Deaf user needed access to a Teldem, and had to pay for calls. The duration of text-based calls was longer than voice calls due to type-time and was therefore expensive. In addition, some Deaf users felt that hearing users would use their phone plugs when the Teldem was absent and they would have to pay for it. Such problems resulted in Deaf users not trusting the Teldem.

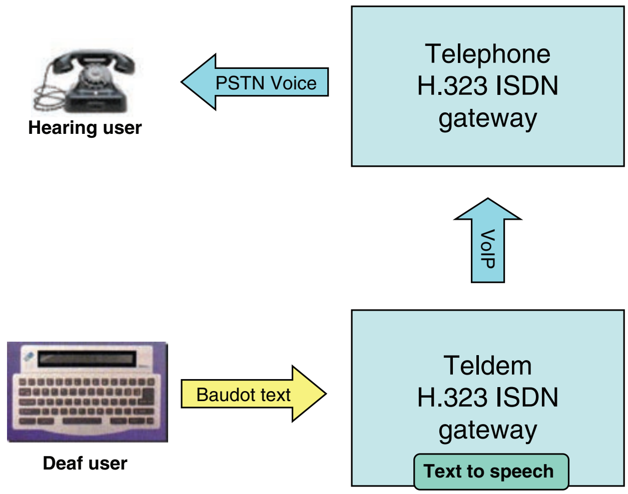
At the time, we considered the Teldem to be the only real-time communication option to give DCCT members an independent means to communicate with anyone, both hearing and Deaf users. One could argue that voice-based synchronous communication could be substituted by asynchronous text-based mechanisms. However, we felt synchronous communication was the only way to reliably ensure that a communicative exchange was actually happening. It was therefore imperative to use synchronous mechanisms. The main problem, as we saw it, was to bridge synchronous communication from Deaf to hearing users. The technical challenge was to convert text on the Teldem to voice on the PSTN, and back again.

Plan Action

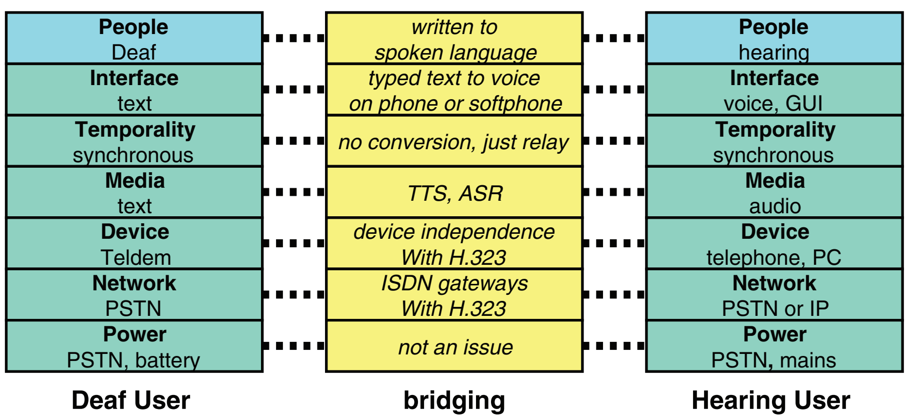
At the time, the Internet offered many text and video-based opportunities for Deaf-friendly communication. The Teldem had no connection to Internet. If the Teldem could interface to the Internet via the PSTN, the Deaf participants' connectivity circle could grow and the prospects of communication independence could increase as well. To work toward this aim, we planned a series of bridges utilising the Teldem (Glaser & Tucker, 2001). The first Deaf telephony bridge would be a system where a human operator relayed communication from a Teldem user to a hearing party on a telephone. We also conceived of various software solutions that offered connectivity between the Teldem and the Internet. A final bridge enabled text and speech to be automatically converted and relayed between text and voice users in real-time.

Implement Action

Directly after presenting the ideas for these bridges at a local telecommunications conference (Glaser & Tucker, 2001), Jason Penton, a Masters student at Rhodes University, offered to code the last bridge because it fitted within his Masters project to create VoIP services with H.323 (Penton, 2003). We called his prototype Telgo323 because it enabled a Teldem to connect to a telephone on the PSTN with ISDN via H.323 (Penton et al., 2002; Tucker et al., 2002). The bridge was implemented in one direction (see figure below). The Deaf user typed text on a Teldem that was converted to speech and delivered as voice to a hearing user. The reverse direction, from speech to text, was beyond the scope of Jason's thesis, and was left as future work. At the time, the Softbridge concept was gestating. The next figure shows the Softbridge stack design of Telgo323 in hindsight.



Telgo323 architecture The Telgo323 prototype relayed text from a Deaf user to voice for a hearing user. The Teldem issued a real-time stream of Baudot-encoded text characters to an H.323 ISDN gateway enhanced with an open source text-to-speech (TTS) engine from Festival. The converted voice was relayed to a telephone with an H.323 gateway. The gateways resided in the IP space where we were able to make software modifications.



Telgo323 Softbridge stack Telgo323's aim was to provide a two-way automated translation bridge between a Deaf user using a Teldem and a hearing person with a telephone or soft phone. Telgo323 only implemented the communication in one direction, using an open source text-to-speech engine called Festival to relay text typed on the Teldem to a handset connected to a PSTN-based PBX or an H.323 endpoint, a soft phone running on a PC.

Evaluate Action

The Telgo323 prototype only worked in one direction from a Deaf user to a hearing user. Because the Teldem's Baudot encoding was non-standard, the struggle of returning text to the Teldem caused us to abandon that task. Another reason to postpone the speech-to-text delivery was that TTS technology worked fairly well at the time, but ASR was difficult to train over the phone. Furthermore, open source ASR tools did not easily recognize South African accented English. Telgo323 was designed to 'plug and play' the ASR and TTS tools so that as technology improved, a new tool could slot into the architecture.

Reflection/Diagnosis

Telgo323 work ceased because Jason completed his Masters (Penton, 2003). In order to test the tool with a Deaf user we needed to convert speech to text, but first we wanted to move the implementation of VoIP to SIP (Handley et al., 1999). SIP was an IETF competitor to the ITU's H.323 protocol family and has since replaced H.323 as the VoIP protocol of choice. We decided the next step would be to port Telgo323 to SIP.

One concern for using VoIP, with the intention of eventually deploying this solution over the Internet instead of just inside our laboratory, was that VoIP in South Africa was illegal (DoC, 1996; DoC, 2001). We felt that the legislation hindered the take up of tools like Telgo323 and kept South African Deaf people even more disadvantaged. We hoped the situation would change and indeed VoIP eventually became legal in 2005 (DoC, 2005).

TelgoSIP
- Ported to SIP
- Still only one way and in lab

TelgoSIP Later in 2002, two interns from the Indian Institute of Technology (IIT), Harsh Vardhan and Nitin Das, came to UWC to port Telgo323 to SIP for their final year project. We still did not engage the Deaf community. However, this prototype is included as an action research cycle for completeness. The table below provides a brief overview.

Plan action

The intention for the interns was only to port Telgo323 to SIP, and not be concerned with implementing the reverse direction (with ASR and text-to-Baudot conversion). We knew this prototype would not be sufficient to trial with a Deaf user, so we did not even plan to do so. We would, however, test the technical performance in the laboratory.

Cycle overview	Description
Timeframe	Mid 2002
Community	N/A
Local champion	N/A
Intermediary	Meryl Glaser (UCT)
Prototype	TelgoSIP
Coded by	Nitin Das and Harsh Vardhan (IIT/UWC)
Supervised by	William Tucker (UCT/UWC)
Technical details	Not published

TelgoSIP cycle overview

Implement action

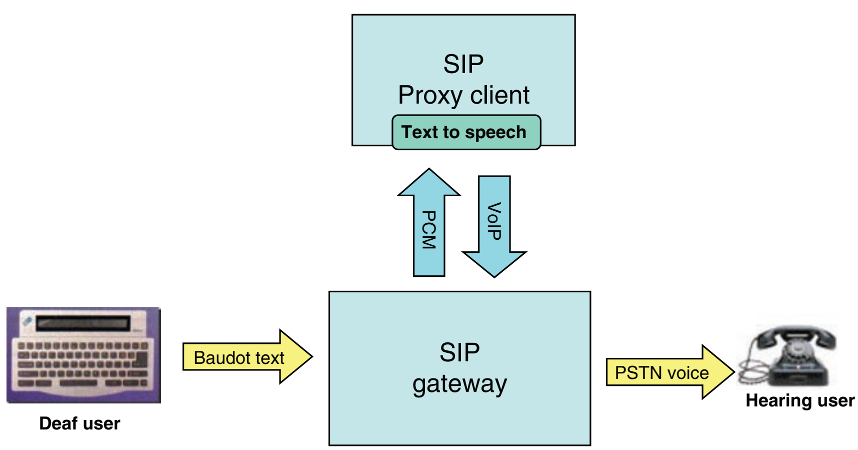
We completed the port from H.323 to SIP and called the new prototype TelgoSIP. TelgoSIP used the VOCAL open source SIP platform from Vovida (http://www.voip-info.org/wiki/view/VOVIDA1SIP). The PSTN gateway was implemented with the VOCAL stack on a QuickNet card. TelgoSIP offered a cleaner solution with modification of a user agent (UA) client instead of modifying gateways as with Telgo323. We still used open source Festival to convert text to speech. The speech was then sent via the QuickNet card using an unmodified VOCAL gateway. The TelgoSIP architecture is shown below. The Softbridge architecture was virtually identical to that of Telgo323.

Evaluation action

Because neither Telgo323 nor TelgoSIP was able to successfully work out the reverse direction, we did not test either prototype with a Deaf person. We brought out another pair of Indian interns to work on that problem in 2003, but they were not able to solve the problem. We therefore aborted what would have been a second TelgoSIP cycle. We had learned that the Teldem was a failure in terms of social take up in South Africa, and was also so difficult to work with on a technical level that further development was unjustified.

Reflection

We decided to abandon the Teldem, but did not abandon the goal for a Deaf-to-hearing relay bridge. The Telgo323 and TelgoSIP prototypes demonstrated a software bridge concept that, when completed in the opposite direction, could enable Deaf people to communicate synchronously with hearing people on the PSTN and Internet. The most obvious bridge was between Deaf and hearing people accomplished by bridging between voice and text modalities. Another bridge was at the network level between the PSTN and the Internet, and consequently between physical devices, e.g. handset, Teldem and soft phone. Another network



TelgoSIP architecture TelgoSIP ported Telgo323 to SIP. TelgoSIP modified a SIP client that acted as a proxy for the PSTN-connected Teldem. Modification at the client level significantly decreased the complexity of the overall design. TelgoSIP did not provide hearing-to-Deaf communication, and therefore was not tested with end-users.

bridge was between high and low bandwidth, because most of the data transport between gateways and/or user agents was text that required much less bandwidth than voice. All of this bridging meant that anybody, hearing or Deaf, with access only to a PSTN line via a handset or Teldem could communicate via IP on the Internet.

Softbridge v1
- Generic modality adaptation
- PC-based CORBA approach

Softbridge v1 While the Telgo prototypes were in development, another MSc student at UCT started working on an approach based on generic adaptors between various disparities. The most obvious adaptation was between text and voice. The preliminary Softbridge abstraction emerged from this effort. This cycle still did not engage the Deaf community, but abandoning the Teldem encouraged IP-based opportunities to build Softbridge reference implementations. The table below provides an overview.

Diagnosis

To perform action research, we needed to test a fully functional bridge with actual end-users. We considered establishing a manual bridge with a human relay operator. Human relay call centres were already in place throughout developed regions, mostly subsidised by government and a respective telco. However, in South Africa, as in many developing regions, the incumbent telco did not provide this service simply because it was perceived not to be able to generate revenue. Our research group did not have the funding or the clout to establish a relay service. The Teldem, produced by Telkom, was difficult for us to integrate into a solution. We had to consider other devices besides the Teldem, and revisited our conceptual bridges (Glaser & Tucker, 2001). The chat interface on the PC appealed because it resembled an Instant Messaging approach.

Cycle overview	Description
Timeframe	Mid 2002
Community	N/A
Local champion	N/A
Intermediary	Meryl Glaser (UCT)
Prototype	Softbridge
Coded by	John Lewis (UCT)
Supervised by	William Tucker (UCT/UWC), Edwin Blake (UCT)
Technical details	(Lewis et al., 2002)

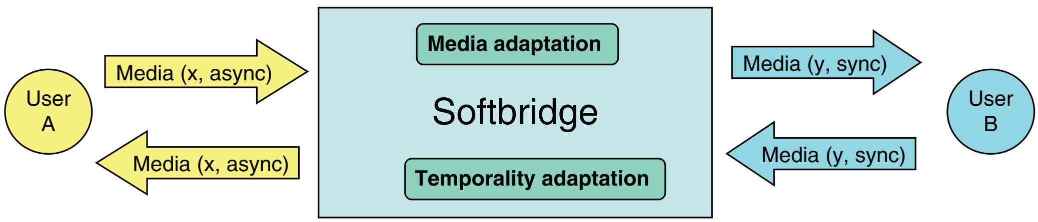
Softbridge v1 cycle overview

Plan action

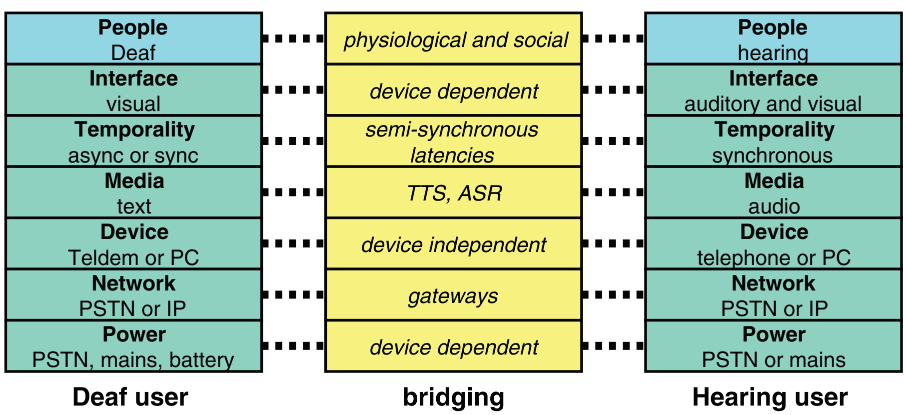
A preliminary Softbridge abstraction shown below emerged from the design specifications of the Telgo and Softbridge prototypes (Lewis et al., 2002). The model concentrated on media and temporality bridging. However, Deaf telephony also required adaptation at other layers in the Softbridge stack (see next figure).

Implement action

A reference implementation was initiated with CORBA (Common Object Request Broker Architecture) as the platform for inter-process communication, but was not completed. Therefore, this cycle emphasised preliminary conception of the Softbridge abstraction. Unlike in the OSI model, the user application itself, not the application network API, was included in the design. The text relay bridge application required media and temporality bridging. Another significant difference from the OSI model was that communication modalities could be converted, or adapted, from one to another based on user capabilities. Thus, it was clear that the user must be included in the Softbridge stack.



Generic Softbridge approach to Deaf telephony The early Softbridge approach adapted various forms of media in and out, e.g. text, voice or video. Softbridge also adapted delivery of media between synchronous and asynchronous temporalities, creating a semi-synchronous form of delivery. The adaptations did have negative quality consequences, however. Text-to-speech and vice versa incurred quality degradation of the message content, and semi-synchronous communication incurred latency in message delivery.



Fully automated text relay Softbridge stack Whereas the Telgo prototypes implemented bridging as part of the Deaf telephony application itself, the initial Softbridge prototype moved the media and temporality adaptation out of the application into middleware. This distinction was significant to the design process in order to hide bridging from the users. Media adaptation could not be hidden so easily. The Deaf user knew the text was being converted to speech, and the hearing user was also painfully aware that a TTS engine was being used. The temporality bridging was more interesting. Both Deaf and hearing users would be aware of the delays incurred due to the TTS and ASR adaptation even though they most probably would not realise the extent to which asynchronous Instant Messaging was being converted to synchronous VoIP.

Evaluate action

Softbridge design implied that bridging synchronous and asynchronous temporalities could produce semi-synchronous exchange. Communication quality in the conventional QoS sense was obviously degraded by the large latencies incurred by media adaptation with TTS and ASR. One research aim was to determine if and how those quality problems could be overcome. While TTS functioned reasonably well, we knew ASR would be problematic. Furthermore, temporality adaptation between asynchronous text and synchronous voice introduced delays into the system, especially for the hearing user. Later, we would instrument the code to record these delays, and also explore ways to measure the quality of the TTS and ASR output.

Diagnosis/Reflection

This cycle produced an initial Softbridge abstraction that was mostly concerned with media and temporality bridging. However, other bridges were also relevant to bridge between Deaf and hearing users with different types of interfaces, devices, networks and their power systems. These considerations would coalesce in the Softbridge models to come. However, the task remained to build a Softbridge prototype that could be tested with the Deaf community.

Softbridge v2
- Jabber async text & voice
- Web service media adapters
- Mixed media clients with .NET
- 1st Deaf user trial in lab

Softbridge v2 The preliminary Softbridge abstraction began to take shape in a second Softbridge reference implementation that successfully provided a bi-directional Deaf telephony system. The action research method also improved during this cycle because of the first engagement of the Deaf community with a Softbridge prototype. We tested the prototype with a single literate Deaf person who provided valuable input about the prototype and more importantly about the social processes surrounding the introduction of such technology. The table below provides an overview.

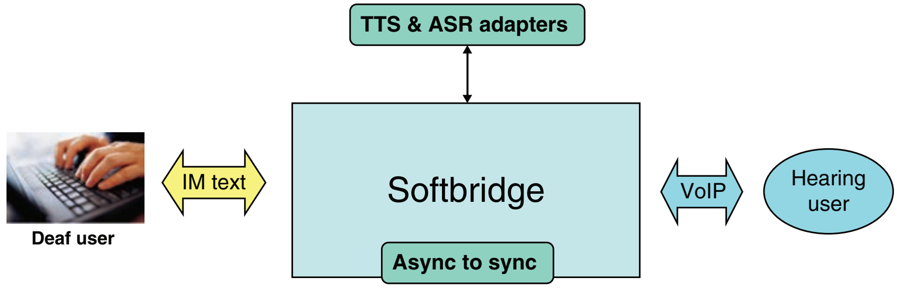
Cycle overview	Description
Timeframe	Late 2002 – mid 2003
Community	DCCT
Local champion	N/A
Intermediary	Meryl Glaser (UCT)
Prototype	Softbridge v2
Coded by	John Lewis (UCT)
Supervised by	William Tucker (UCT/UWC), Edwin Blake (UCT)
Technical details	(Lewis et al., 2003; Tucker, 2003; Tucker et al., 2003a, 2003b)

Softbridge v2 cycle overview

Plan Action

The plan was to provide Deaf-to-hearing communication in both directions. Leveraging the ease of IP-based tools, we planned to employ an IM interface on a PC for the Deaf user. The system would convert IM text to voice for the hearing user on some form of audio device, and convert the hearing user's spoken voice back to text for delivery to the Deaf user.

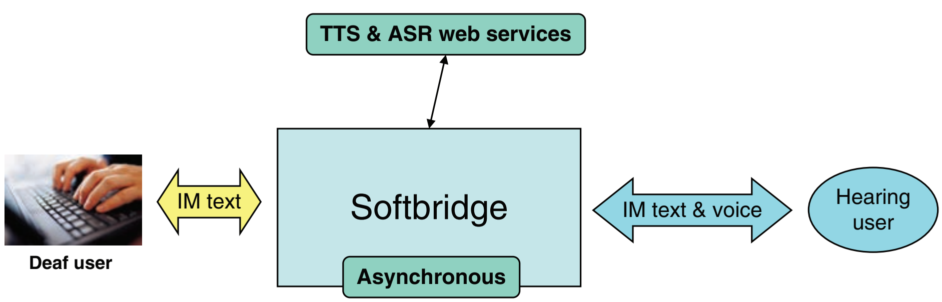
We designed a bi-directional prototype based on the early stages of the evolving Softbridge concept (see below). We named the prototype Softbridge, but this should not be confused with the Softbridge abstraction. Here, the Softbridge prototype was an instance of the Softbridge abstraction (as were all subsequent Deaf telephony prototypes in one form or another).



Instant Messaging-based Softbridge architecture The Softbridge prototype would replace the Teldem device with an Instant Messaging client running on a PC. The Softbridge would automatically adapt between voice and text, and also between asynchronous Instant Messaging for the Deaf user and synchronous VoIP for the hearing user. The adaptation between text and voice would be implemented with web services.

Implement Action

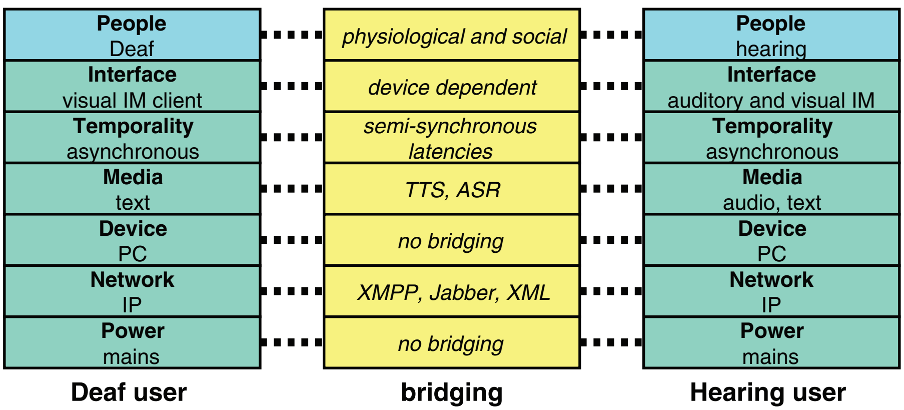
The actual architecture for the implemented prototype is shown below and was described by Lewis et al., (2003). The system architecture was based on a generic Softbridge core with clients specifically built for Deaf telephony. A Jabber-based IM client coded with .Net provided text in and out for the Deaf user. The hearing user also used a Jabber client, but for speech. We also built a hearing client that had both voice and text so the hearing user could see what the Deaf user typed in addition to hearing the TTS-generated speech. We also wanted the hearing user to be able to type messages to the Deaf user when the ASR failed to deliver intelligible text. The TTS and ASR engines were wrapped as loosely coupled web services, thus making them very easy to plug and play.



Softbridge v2 architecture The actual architecture differed slightly from the planned approach shown earlier. Both text and voice transport were implemented with an asynchronous protocol. That meant no adaptation to synchronous VoIP. We reasoned this was acceptable because there were already large delays introduced by TTS and ASR adaptation. Those adapters were loosely coupled web services. We built a variety of clients for the hearing user. For example, one had voice only and another had both voice and text so the hearing user could see exactly what the Deaf user typed.

Instead of using real-time VoIP, a speech clip of maximum 2MB was packaged into XML (eXtensible Mark-up Language) messages to and from a modified Joey server (an open source Jabber server). The modified Joey server effectively became a Softbridge server. The speech XML packets were delivered with XMPP, Jabber's asynchronous protocol. Therefore, no temporality adaptation was required or performed. The intention was to deliver the voice messages as quickly as possible to appear semi-synchronous. However, the TTS and ASR media adaptation in both directions introduced additional delay and also often distorted the message content.

We tested the prototype with a Deaf user who was somewhat unique in that he was PC literate and also had some experience with research studies (Tucker et al., 2003b). We built several clients to experiment with different types of bridging, mostly at the media layer (see below). The Deaf user client was text in/text out. The hearing client allowed us to pick and choose media capabilities. We experimented with the following combinations: text & TTS in/text out, TTS in/text out or TTS in/ASR & text out. The reason for the combinations was to use text to clarify the output of the TTS and ASR. We began the experiment with instructions interpreted to SASL, then used the tool itself to convey instructions to the Deaf user.



Softbridge v2 Softbridge stack For the Softbridge v2 prototype, bridging was not necessary at the lower three Softbridge stack layers. We used this prototype and several client interfaces to conduct an experiment with a Deaf user.

Evaluate Action

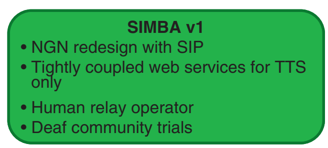
We found that the Deaf user treated the IM client like a Teldem. He was not aware that characters were not being sent one at a time, but rather in page mode, or one message at a time. He also used the GA terminator as if in half-duplex mode. He informed us that Deaf users would type English slowly, and likely with SASL grammar. He gave us some examples, and of course, the TTS engine could only pass on what it received. On the hearing side, TTS messages arrived unexpectedly. The natural rhythmic exchanges in voice conversation were disrupted by the delays caused by typing time and TTS conversion.

Reflection

Our intermediary had long urged us to test the prototype with real Deaf users; that building solutions in the laboratory was pointless. The compromise was to test a solution in the laboratory with a Deaf user. We had emphasised the Deaf user interface, but learned that we also had to address the needs of hearing users. A case in point was the way that the TTS speech appeared without warning. The Deaf user had a persistent textual representation of the conversation in the IM interface. However, a hearing user with an audio-only interface was at a disadvantage not being able to see the conversation in order to rescue the misunderstandings due to spelling/grammar mistakes and poor ASR conversion.

The MSc student had devoted a substantial amount of time to the technical implementation of the Softbridge prototype and its clients. He managed to bridge voice from the Jabber-based platform to an H.323 soft phone client. Unfortunately, the effort ceased prematurely before an intended H.323 gateway was provided to connect the system to the PSTN. Implementation interruptions of this sort had happened several times already because the lead programmer was an MSc student who either graduated or terminated their studies.

5.2 Community based action research (2004-2007)



SIMBA v1 This cycle continued the firming up of the Softbridge abstraction, and instigated a completely different reference implementation. SIMBA v1 provided bi-directional Deaf-to-hearing communication to the PSTN with a human relay operator. This cycle produced the first full-scale engagement with the Deaf community at the Bastion of the Deaf in Newlands. The DCCT NGO abetted the intervention. We visited the Bastion twice weekly where SIMBA was tested with a number of Deaf people who participated in a PC literacy course. The table below provides an overview.

Cycle overview	Description
Timeframe	Mid 2004 – mid 2005
Community	DCCT members
Local champion	Stephen Lombard (DCCT)
Intermediary	Meryl Glaser (UCT), DCCT staff
Prototype	SIMBA v1
Coded by	Sun Tao (UWC)
Supervised by	William Tucker (UCT/UWC)
Technical details	(Blake & Tucker, 2004; Glaser & Tucker, 2004; Glaser et al., 2004, 2005; Sun & Tucker, 2004; Tucker, 2004; Tucker et al., 2004)

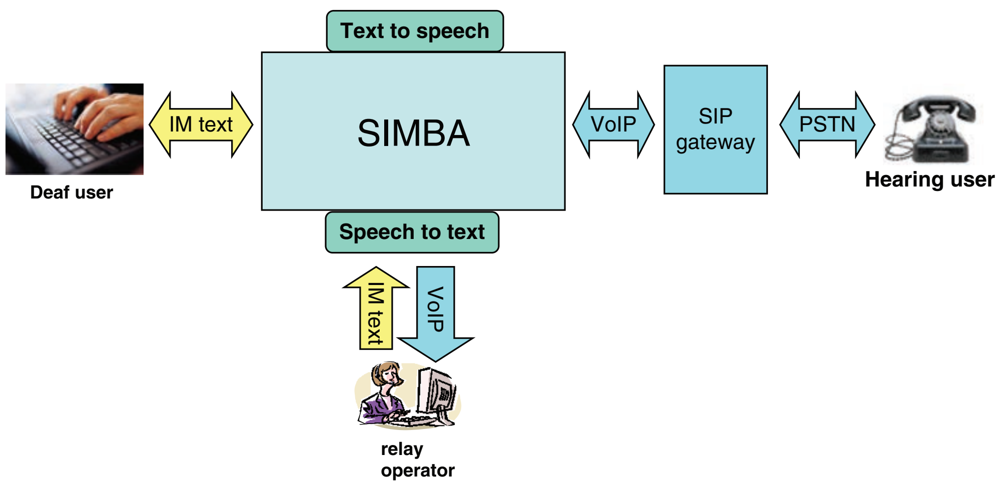
SIMBA v1 cycle overview

Diagnosis

Several concerns emerged from the initial Softbridge trial. Firstly, we had to trial the technology with more users. However, Deaf users from the DCCT community would struggle with both PC and text literacy, causing problems for both Deaf and hearing users. Secondly, ASR was not adequate for generalised recognition. Lastly, we needed to put more students onto the project to avoid relying on only one programmer.

Plan Action

To begin to address these issues, we planned a small PC lab for the Bastion with Internet connectivity. We also needed a PC literacy course specifically dedicated to the target Deaf community so they could learn to use those computers. We also assigned parts of the project to several more students, one of who came up with the next prototype architecture shown below.

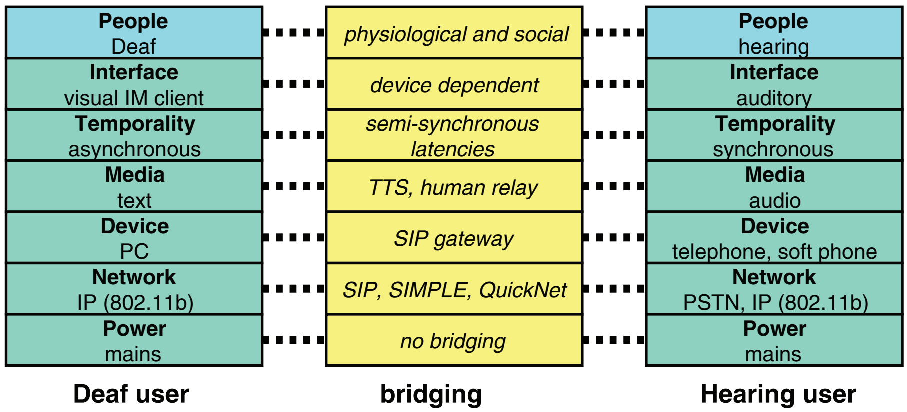


SIMBA v1 architecture SIMBA v1 was similar in design to the previous prototype, but had several key differences. SIMBA had tightly coupled web-services for TTS and ASR, replaced ASR with a human relay operator (though still wrapped as a web service), provided IM with SIP instead of Jabber and implemented VoIP with SIP instead of H.323. SIMBA was intended from the start to breakout to the PSTN.

Implement Action

In mid-2004, Glaser et al. (2005) trained 20 Deaf people on basic PC literacy skills at the Bastion using the newly installed computer lab equipped with ADSL broadband connectivity. The course prepared participants for the trial of the next text relay prototype (Glaser et al., 2004). We spent two afternoons a week helping them get accustomed to typing, email and Internet. We also hired a Deaf assistant (a part-time DCCT staff member) to work with Deaf users, and an interpreter to help us interview the Deaf participants.

The SIMBA prototype enabled communication between a Deaf person using an IM client on a PC situated at the Bastion and a hearing person using a phone or cell phone. SIMBA had several significant differences from the prior Softbridge prototype. SIMBA bridged asynchronous IM with real-time voice with both VoIP and the PSTN. The Softbridge layers are shown below.



SIMBA v1 Softbridge stack SIMBA v1 provided the next reference implementation of the Softbridge abstraction. Several modifications were based on previous experience. A human relay operator replaced the ASR with an interface wrapped as a web service so it could be easily replaced at a later date. SIMBA also bridged asynchronous IM text to fully synchronous VoIP with SIP and a SIP gateway to the PSTN. Hearing users could use a variety of devices: telephone, cell phone and PC-based soft phone.

Evaluate Action

Between the PC literacy class in July 2004 and the introduction of SIMBA in December, an average of six Deaf people came every Wednesday, and five people every Thursday to the PC lab at the Bastion. We had installed five PCs with video cameras. When we were not there, the PCs were covered with a cloth. This indicated that DCCT staff wanted to take care of the machines. However, no one would use them while they were covered. Demonstrating more dedication to the project, DCCT built a dividing wall in the office to create a distinct PC lab within their office space.

In 2005, we hired one of the PC literacy class participants (a part-time DCCT staff member) to keep the lab open during the day four days per week. Attendance averaged three people per day. We still visited twice weekly for two-hour research sessions, but the attendance during this cycle dropped off to as low as two people per session. We began exposing Deaf participants to SIMBA during these sessions.

We asked all potential SIMBA users, Deaf and hearing, to sign research consent forms. This was a long and drawn out process. Deaf people signed fifteen consent forms, and hearing people completed fourteen. Before initiating a SIMBA call, we would call the hearing recipient to inform them what was happening. Many of the initial SIMBA calls were hindered by the fact that the hearing people called were working at the time or the system malfunctioned or crashed.

We instrumented SIMBA to collect usage statistics and record conversation transcripts. We also added instrumentation to measure the delays in the various stages of the Deaf-to-hearing communication process. However, so few calls were made, the usage statistics were not useful. The only successful SIMBA calls were made between one of the DCCT staff and a hearing social worker assigned to DCCT. One of those calls was 45 minutes long, but most of them were much shorter. SIMBA transcripts were used to determine that hearing people could indeed understand the voice synthesised from Deaf text despite poor spelling and grammar (Zulu and Le Roux, 2005).

Reflection/Diagnosis
Engagement with more Deaf participants during the cycle led to the recognition of several significant challenges. Most importantly, both textual and PC literacy was evidently lacking. We began to address these deficiencies with training and follow-up sessions (Glaser et al., 2004; Glaser et al., 2005), yet both text and the PCs continued to intimidate the Deaf users. Typing skills were so poor that the scheduled research sessions consisted mostly of practice with a typing tutor. During the weekly visits in 2004, we observed that another reason Deaf participants were not using Internet-based communication tools was that they had no one to email with, and certainly no one to IM with as their friends and relatives did not have access to ICT at home or at work.

A number of technical issues arose from the SIMBA prototype. This was to be expected because it was the first used outside of the laboratory. There were niggling user interface issues such as faulty scrolling and presence that were easily fixed. More serious problems involved continuous rebooting of the system components in order to get the system ready to make a call.

The research protocol also hindered usage of SIMBA because we insisted that all users, Deaf and hearing, sign consent forms. This proved difficult because we had to rely on the Deaf participants to deliver and return the signed consent forms from their hearing co-participants. They often did not understand that we wanted them to do this. In the end, we dropped the requirement for hearing participants since we were most interested in building solutions for Deaf people. We were also not recording the hearing user's voice, although we did record the relayed text. It was telling that the most successful SIMBA experiences were performed at the DCCT premises between two staff members (one Deaf, one hearing).

SIMBA v2 and NIMBA
- Audio isTyping
- Guaranteed delivery

SIMBA v2 We hired the lead student from the previous cycle, Sun Tao, to code for the project full-time after he finished his MSc. His first task was to fix bugs and to provide awareness features for both Deaf and hearing users. This resulted in an innovative audio isTyping awareness feature that let the hearing user 'hear' when the Deaf user was typing. Another MSc student wrapped SIMBA with a guaranteed delivery framework for both synchronous and asynchronous communication. SIMBA trials continued on a weekly basis, with very little change in attendance and usage. The table below provides an overview.

Cycle overview	Description
Timeframe	Late 2005
Community	DCCT members
Local champion	Stephen Lombard (DCCT)
Intermediary	Meryl Glaser (UCT), DCCT staff
Prototype	SIMBA v2
Coded by	Sun Tao (UWC), Elroy Julius (UWC)
Supervised by	William Tucker (UCT/UWC)
Technical details	(Hersh & Tucker, 2005; Julius, 2006; Julius & Tucker, 2005; Sun, 2005)

SIMBA v2 cycle overview

Plan Action

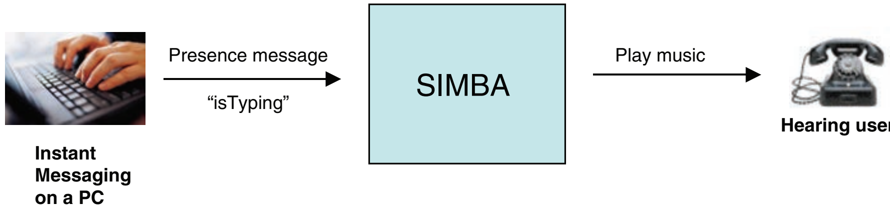
This cycle's main focus was to fix the bugs in SIMBA v1 and make SIMBA v2 more usable and reliable. We had noted that the long latencies made communication difficult for hearing users by upsetting expected voice conversation rhythms. We planned to leverage IM awareness and presence mechanisms to deal with such macro latency. A simple message on the IM client GUI (graphical user interface) would let the Deaf person know that someone was speaking, and furthermore, that speech was being converted to text. We wanted to provide a similar awareness feature for the hearing user, such that the hearing person would know that text was being typed and/or converted to voice. We would play a tone to let the hearing user know that this was happening. We called this 'audio isTyping'.

Another MSc student joined the project. His research topic was to explore guaranteed delivery of semi-synchronous messaging (Julius & Tucker, 2005). The purpose of his study was to demonstrate to the Deaf user that SIMBA could guarantee delivery of messages where SMS could not.

Since we had poor participation during the weekly sessions in the previous cycle, we decided to concentrate on the DCCT staff members who were working at the Bastion. In order to do that, we had to expand the wireless network and put PCs on their office desks.

Implement Action

We identified and fixed many SIMBA bugs, and added a number of features during this cycle: configuration parameters for gateways, dynamic instead of fixed IP addresses, handled a busy signal, enabled the relay operator to terminate a call, parameterised the TTS engine via the SIMBA interface and recovered when the telephone hung up. One particular problem with the relay operator's audio was difficult to fix: the operator heard the outgoing result of the Deaf user's TTS. The most significant feature addition was isTyping for the Deaf user, and audio isTyping for the hearing user (see below). The overall SIMBA architecture remained as depicted above.



SIMBA v2 audio isTyping awareness feature The audio isTyping feature caused SIMBA to play music for the hearing user when the Deaf user was typing text. SIMBA also provided awareness for the Deaf user. A message was displayed on the Deaf user's IM client when the hearing user was speaking.

An offshoot of SIMBA, built with the Narada brokering facility from Indiana University (www.naradabrokering.org) was implemented as NIMBA (Julius, 2006). NIMBA provided guaranteed delivery of real-time components with forward error correction and of store-and-forward components with Narada.

Several old PCs had been donated to DCCT and we put these in the staff offices. We extended the WiFi network to include staff offices with a second AP with a strong directional antenna (borrowed from our rural telehealth project) to get through the thick brick walls.

Evaluate Action

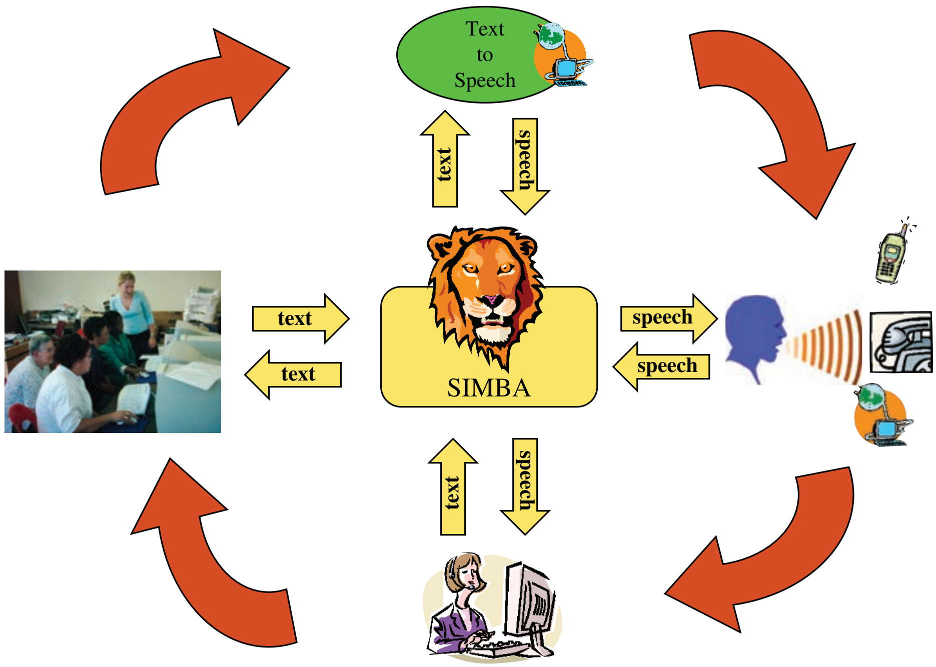
NIMBA experiments were conducted with several regular attendees, but we had little success getting them to use either NIMBA or SIMBA. It continued to be very difficult to reach hearing co-participants. There was only one instance when a Deaf person asked us to set up a SIMBA call for them. Instead, we were continually asking them to make a SIMBA call. One person noted that if a Deaf person wanted to contact a hearing person, they would just use SMS.

One technical problem with SIMBA was that its FOSS TTS engine, FreeTTS, stopped processing text after the first full stop (period) in the Deaf user's message. Thus, when a Deaf user typed a long message with multiple sentences, only the first sentence would be sent to the hearing user. Therefore, we encouraged the Deaf participants to use short single sentences. FreeTTS also did not intonate punctuation like other TTS engines, so the result appeared quite mechanical and synthetic to hearing users.

Reflection/Diagnosis

During focus group sessions, DCCT staff members identified several inhibiting factors regarding the poor take-up of SIMBA. Deaf users could not use the system when we were not there. Firstly, this was due to the continued problems with operating SIMBA. Secondly, SIMBA was closely associated with our presence. Because of the poor take-up, we did not hire a relay operator so one was not always available. The end result was that Deaf users could not use the system any time they wanted.

Consent forms also inhibited take-up. Very few people brought in consent forms for hearing users. At first, the Deaf users completely misunderstood that the consent form was supposed to be signed and brought back because they had had little experience with research and research protocol. They often brought back the written project information sheet instead. They also had difficulty understanding the text. We made graphical sketches of the system, and then put large posters in the Bastion (see below). Those helped explain the project to the Deaf users better than the written text.



Visual information sheet for the SIMBA system The Deaf users had difficulty with the written information sheet for the project. We drew this graphical depiction of the system for them, and also placed a large poster in the PC room at the Bastion. We would write subsequent information sheets (for other prototypes) in point form to make it easier for an interpreter to translate into SASL.

Our operation hours were also awkward. They may have been convenient for the Deaf users, but were not convenient for the hearing users. Thursday evening sessions were problematic because many hearing co-participants were Muslim, and did not want to be disturbed at dusk during prayer time. The other time slot was Friday morning when hearing participants were working.

Some significant non-research events also occurred during this cycle. The PC lab assistant left to have a child and was replaced by someone else who became difficult and ceased working for the project. Thus, we realised we should employ more than one assistant in case we had similar problems in the future. Our initial intermediary resigned her post at UCT and began working for another Deaf NGO called SLED (Sign Language Education and Development, www.sled.org.za). However, she remained active with DCCT, and with our project.

Most significantly during this cycle, DCCT said they would budget for ADSL the following year. This demonstrated buy-in to the project, and acknowledged the importance of having the computers at the Bastion. We thus came to view DCCT also as intermediaries.

We learned that the Deaf people in this community were not accustomed to calling or sending an SMS to 'any sector'. They were used to communicating within their tight knit Deaf circle, and felt cut off from other sectors. When asked what they wanted from a SIMBA-like system, some of the Deaf people replied that they wanted to use SASL instead of text and one recommended access to SIMBA with SMS in addition to IM. Since we were still trying to automate as much of the system as possible, we opted to follow up on the latter request.

SIMBA v3
- SMS interface added for Deaf user
- Added Asterisk & Digium

SIMBA v3 A third version of SIMBA provided an SMS interface to the Deaf user. SIMBA v3 trials continued on a weekly basis, yet also failed to attract users. However, the project saw significantly increased use of the PCs in the Bastion. The table below provides an overview.

Cycle overview	Description
Timeframe	Early 2006
Community	DCCT members
Local champion	Stephen Lombard (DCCT)
Intermediary	Meryl Glaser (SLED), DCCT staff
Prototype	SIMBA v3
Coded by	Sun Tao (UWC)
Supervised by	William Tucker (UCT/UWC)
Technical details	None published

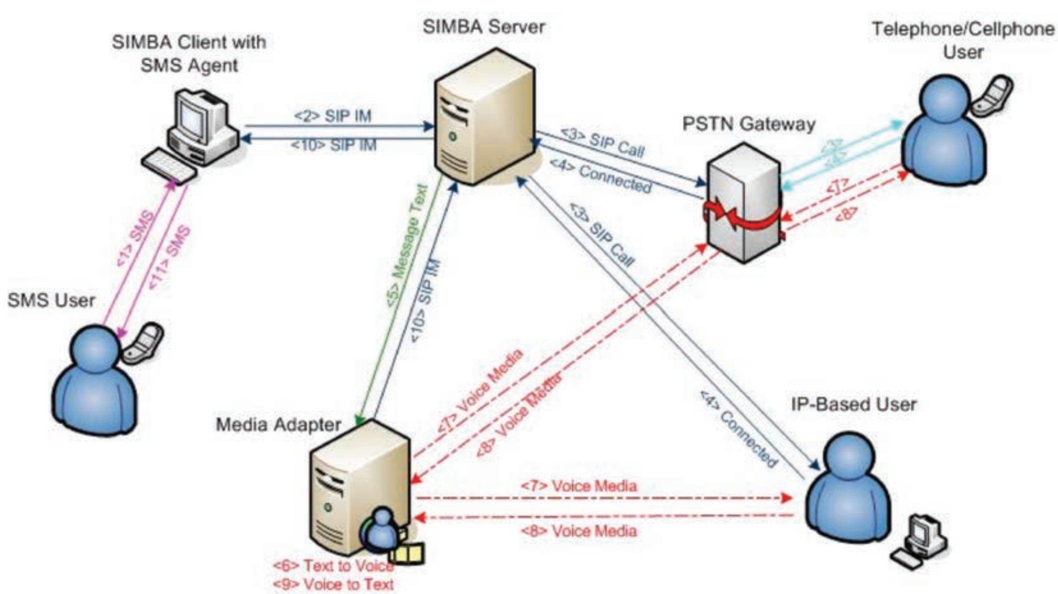
SIMBA v3 cycle overview

Plan Action
A brief overview of the technical design of the SMS interface is presented in the transaction diagrams below. SIMBA v3 was particularly interesting in that it involved bridging at all seven Softbridge stack layers (see below).

Implement Action
An SMS agent was integrated into a SIMBA client. The SIMBA server was not changed. The SMS agent used a GPRS card to send/receive an SMS. To initiate a SIMBA call, the Deaf user sent a specially formatted SMS to the SMS agent. From then on, the user sent and received SMS as normal, with SIMBA performing relay to a telephone.

Deaf user	bridging	Hearing user
People Deaf	*physiological and social*	**People** hearing
Interface SMS client	*device dependent*	**Interface** auditory
Temporality asynchronous	*semi-synchronous*	**Temporality** synchronous
Media text	*TTS, human relay*	**Media** Audio
Device cell phone	*message conversion*	**Device** telephone, PC, cell phone
Network GSM	*GSM and SIP gateways*	**Network** PSTN, IP, GSM
Power handset battery	*virtual electricity bridges*	**Power** PSTN, mains, battery

SIMBA v3 Softbridge stack SIMBA v3 was particularly interesting because it involved bridging at all seven Softbridge layers. The SMS interface for the Deaf user entailed differences in power provision, network access, end-user devices, text and voice media, synchronous and asynchronous temporalities, user interfaces and of course between Deaf and hearing people.



SMS interface to SIMBA v3 This diagram shows the sequential flow required to add the SMS interface to SIMBA. The arcs are defined below.

1 SMS user sends an SMS to SMS Agent. The message looks like "*0722032817* how are you?". The content between the asterisks is the called user's number or name.
2 SMS Agent extracts the message from SMS and formats a SIP IM for the SIMBA Server. The SIP IM looks like "sip: 0722032817@Softbridge.org".
3 SIMBA Server receives the message and sets up a SIP call to the called user, either via a PSTN Gateway for a telephone/cell phone or directly if it is an IP-based user.
4 Called user answers the call and a connection is set up.
5 SIMBA Server sends the IM message text to Media Adapter.
6 Media Adapter converts text to voice.
7 Media Adapter sends voice stream to called user.
8 Voice stream is sent from called user to Media Adapter.
9 Media Adapter converts voice to text via Relay Operator.
10 Media Adapter packs text to SIP IM and sends it to SMS Agent via SIMBA Server
11 SMS receives SIP IM, gets text message, packs to SMS and sends to a Deaf user.

Flow diagram for SMS interface to SIMBA v3

Evaluate Action

Experimentation with SIMBA v3 occurred as in earlier cycles, in twice-weekly research sessions. Participation continued to be sparse, despite increased daily attendance for the open lab. As expected, the SMS interface exhibited long latencies due to 'tap' time. Unfortunately, take up did not improve and we conceded that SIMBA was not going to be taken up as a service by this community.

Reflection

The SMS interface and isTyping features were not enough to boost SIMBA usage. In hindsight, both were technically interesting ideas, but added little value for potential users. We came to realise that addressing perceived problems with yet more technical bells and whistles was not going to improve take-up. The problems were deeper, and more social in nature. We were told by DCCT leaders that perhaps the Deaf community was so close knit that the members really had no desire to 'talk' to people outside the community. They were more interested in ICT that enabled them to communicate with one another, like SMS, email, IM and especially video conferencing. This was clearly evident by attendance during the week at the PC lab. With two lab assistants, the open days for the lab were increased to six days/week, and overall attendance increased dramatically. While technical research with SIMBA was floundering, the lab was being used in record numbers two years after we had introduced the PCs.

Related to the sentiment above that our technical concerns were overshadowed by the social factors, we contemplated the efficacy of the informed participation approach as described by Hersh and Tucker (2005). At the time, our main concern was to explore QoC and learn how macro latencies could be overcome to provide a still useable communication platform. We quite openly discussed delay issues with participants. It might have been that they may were confused about the purpose of the project: was it PCs, Deaf telephony, or delay?

With a full-time programmer on the project, we were able to more quickly address bugs and enhancements. Unfortunately, the lead programmer immigrated to Canada and we replaced him with another MSc graduate.

Deaf Chat
- Real-time text chat similar to Teldem, but multi-user and PC-based
- Deaf users like it & use it
- Standalone and web clients with SIP

DeafChat We next built a real-time text chat system that sent characters to chat participants as they were typed instead of waiting for the terminating new-line. DeafChat proved to be very popular during the weekly research sessions. The Deaf users actively participated in the design of the tool by offering feedback. Since the programmer was not a student, we were able to react quickly to their suggestions. The table below provides an overview.

Cycle overview	Description
Timeframe	Mid-late 2006
Community	DCCT members
Local champion	Stephen Lombard (DCCT)
Intermediary	Meryl Glaser (SLED), DCCT staff
Prototype	DeafChat
Coded by	Yanhao Wu (UWC)
Supervised by	William Tucker (UCT/UWC)
Technical details	None published

DeafChat cycle overview

Diagnosis
It appeared that the Deaf people were more interested in communicating with each other than with hearing people. We had introduced them to IM systems like MSN, Skype and AIM, but rarely saw them using those tools. Many of the Deaf participants had prior experience with the Teldem, even if they did not own one or use one. We reasoned that a real-time text chat tool, similar to the Teldem in synchrony, might appeal to Deaf people.

Plan Action
We designed a real-time text tool that transmitted characters in real-time, similar to the Teldem. Unlike the Teldem, however, this tool would support multiple participants, have a PC GUI interface and identify users. IM clients typically transmitted in 'page mode', meaning that text was transmitted in chunks defined by the user hitting the Enter key (new-line). The Teldem operated in 'session mode', sending one character at a time. Thus, the plan was to imbue an IM client with Teldem-like synchrony since we could not add IM functionality to the Teldem.

Implement Action
We built two versions of DeafChat, both of which were client-server in nature. The first version was built with SIP, using the MESSAGE mechanism to pass one character at a time to the server that would then broadcast the characters to all connected clients. The clients managed the screen, relegating the character to the appropriate position. A sample interface is shown

below. Based on user feedback, we colour coded users' text and improved the interface. We subsequently moved the entire application to a Java-coded web-based client to remove the necessity of having to install and upgrade software.

DeafChat interface The standalone SIP implementation of DeafChat had implicit isTyping. As a user typed, characters were sent to all other chat participants and positioned according to that user's current message. After pressing Enter, a 'GA' token appeared to indicate: "I am finished typing now". GA was a holdover from the Teldem. Multiple users could type simultaneously. Any characters typed in a local input box appeared both in the local output box and any connected client's output box in the correct position in real-time.

Evaluate Action

DeafChat was well received by the regular research session participants in a way we had not experienced before. During this cycle, the whole group frequently used DeafChat during the research sessions. When new people came in, they often asked to get on it, too. The tool was also used by the weekly literacy course. Sometimes, they would already be involved in a group chat before we arrived. They also did not mind us participating in their chats, which mostly consisted of poking fun at one another, and at us. DeafChat was rather basic. Login was not authenticated and the application was limited to the local area network, but these issues did not matter to the Deaf users.

Reflection

There were several innovative technical features of the tool. At the network layer, the tool appropriated SIP messages to deliver text characters in real-time. SIP was designed for real-time voice and video (Handley et al., 1999), and only provided for asynchronous text with SIMPLE (Campbell et al., 2002). At the interface layer, DeafChat provided isTyping awareness implicitly. A user could always tell what the other chat participants were doing (or typing) based on a quick visual scan of the screen. DeafChat was deployed on PCs at the Bastion, yet the web-based client interface made it possible to port the DeafChat to cell phones. However, that would still require a GPRS, 3G or WiFi data connection.

After struggling for so long with the SIMBA prototypes, we were pleasantly surprised by the instant popularity of DeafChat. However, that popularity was short-lived. After the novelty wore off and Yanhao Wu (the programmer) left the project, DeafChat retired into the same disuse that befell prior prototypes. In spite of that, there were several instructive explanations for its brief success. Unlike SIMBA, DeafChat was fundamentally a Deaf-to-Deaf tool. DeafChat was modelled on Teldem-like modalities familiar to the participants, and enabled them to feel comfortable with textual content amongst themselves, much as with SMS. It was notable that the participants were also comfortable with the researchers participating in the chats. The participants had acquired a solid base of computer literacy via the previous prototype trials. In addition, most of them were also participants in an on-going English literacy course that was co-scheduled with the research visits. Thus, the Deaf participants felt more confident using both ICT and English text. Most importantly, participants used DeafChat with each other because they all had the same degree of technology access in the lab during the research sessions. This also explained why DeafChat could not achieve larger penetration into the Deaf community; a user had to be physically in the lab to use it. When the circle was closed and small, DeafChat usage was encouraging. Yet outside the circle, potential users were sidelined by lack of access to technology.

Deaf Video Chat v1
- Semi-synchronous video
- Real-time video
- User trials of codecs and transport protocols

DeafVideoChat We exposed Deaf users to several off-the-shelf video IM tools to respond to the need for SASL communication. None of the common IM video tools appeared to support sign language conversation with webcams, even relatively high-end webcams with large amounts of P2P bandwidth on a local network. Thus, we conducted a preliminary investigation into asynchronous video in 2006 with an Honours (4th year) project. We continued the project in 2007-8 as an MSc project. We also began to explore an innovative gesture recognition interface to the asynchronous video prototype. The table below provides an overview of the cycle.

Cycle overview	Description
Timeframe	2006-2008
Community	DCCT, SLED
Local champion	Richard Pelton (DCCT)
Intermediary	Meryl Glaser (SLED), DCCT staff
Prototype	DeafVideoChat
Coded by	Zhenyu Ma (UWC), Russel Joffe (UCT), Tshifhiwa Ramuhaheli (UCT)
Supervised by	William Tucker (UCT/UWC), Edwin Blake (UCT)
Technical details	(Ma & Tucker, 2007, 2008)

DeafVideoChat cycle overview

Diagnosis
A primary need expressed by Deaf participants was to communicate in SASL. We had temporary success with DeafChat, but it did not support video. We reasoned that we should abandon the 'do-it-yourself' approach characterised by the SIMBA cycles and expose the Deaf users to off-the-shelf video tools in order to learn about their advantages and disadvantages.

Plan Action
We would expose the Deaf participants to off-the-shelf free video tools available online. The aim was to familiarise them with what was available, encourage them to use the tools and to learn how they performed with respect to Deaf needs. While participants were exploring synchronous video, we would also investigate asynchronous video to provide higher quality.

Implement Action
DCCT participants dismissed Skype and other common video chat tools because of poor video quality for sign language. The tools appeared to be designed for hearing users, prioritising voice over video quality. Thus, sign language communication was blurry and jerky. This prompted us to implement an asynchronous video tool, herein called DeafVideoChat. It was a simple tool with two video windows as shown below. The outgoing recorded video could be replayed to check for correctness, and the incoming video could also be replayed. Exploratory experiments were carried out with various compression techniques, and also with several transmission techniques in order to determine an optimal combination (Ma & Tucker, 2007). Further experimentation at an MSc level resulted in the integration and optimal configuration of the x264 codec into the system (Ma & Tucker, 2008). We also began work on a gesture recognition interface. This should not be confused with sign language recognition. The point of the gesture recognition interface was to provide gestured control of the asynchronous video tool, e.g. to start or stop recording.

We upgraded the webcams in DCCT twice over the course of this cycle. After frustration with tools like MSN and Skype, some DCCT users started using Camfrog (www.camfrog.com) as their preferred video tool for remote sign language communication. We bought Camfrog Pro licenses to enable the use of full screen video and some other features. Some DCCT users took advantage of the social networking that Camfrog offered for Deaf communities around the world. Camfrog's lack of privacy controls made it cumbersome for users who did not enjoy open access from the global Internet community. DCCT users also used the tools to chat to the SLED NGO, and the researchers.

Evaluate Action
Iterative revisions of DeafVideoChat configured the x264 video codec to improve sign language compaction for store-and-forward transmission. One of the Deaf participants reported that it was the first time he had seen clear enough video to understand the sign language. However, Deaf participants did not take to DeafVideoChat as with DeafChat. Firstly, they were more interested in real-time communication. Secondly, the issues regarding the size of the connectivity circle were still relevant. No Deaf people in the potential connectivity circle physically outside the Bastion could communicate with the tool.

DeafVideoChat interface The Capture window on the left was for user1 to capture a SASL video message to send to user2. The Playing window on the right was for user1 to play and replay the last SASL message received from user2. Information messages were displayed in English in the bottom right-hand corner.

On the real-time front, even though Camfrog Pro enabled full screen video, the Deaf users chose a smaller screen size in order to increase video quality. We noted that several frequent Camfrog users with DCCT tended to use Camfrog for its community features. Meanwhile, we were aware that SLED (another Deaf NGO nearby) actively used Camfrog to conduct its daily business because they had two offices, one in Cape Town and one in Johannesburg. SLED users had deemed Camfrog to offer superior quality to Skype.

A prototype of the gesture recognition interface was shown to Deaf users with favourable responses. Unfortunately, the gesture recognition interface project was temporarily halted then restarted as the responsibility passed from one MSc to another.

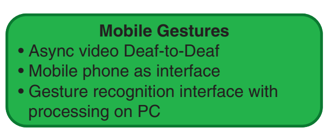
Reflection

DeafVideoChat clearly offered superior video quality with respect to sign language comprehension. Still, Deaf users who actually used a remote video tool rather chose Camfrog. It should be noted that very few of the participants used a video communication tool outside the research sessions. There were only a couple of regular DCCT video users. SLED, on the other hand, adopted Camfrog as a part of their everyday business conduct. There were several explanations why regular users preferred the lower quality tool, and why so few DCCT participants used the tool.

Camfrog was synchronous and was therefore easier to use than the asynchronous interface that required numerous button presses to record, receive and send video. Camfrog also had the advantage of being clearly associated with Deaf users on an international basis. Camfrog provided established chat rooms for users from all over the world. Deaf users related to Camfrog as a SASL tool whereas Skype was a text tool even though it also supported video. DeafVideoChat, on the other hand, was a research tool clearly associated with our weekly visits. Camfrog's perceived superior quality may have been linked to these social issues rather than purely technical ones. We cannot say for sure because we could not perform automated objective video quality tests on Camfrog or Skype as we could with DeafVideoChat because the internals of the web tools were not accessible via open source. On a related issue of users adopting a lesser quality tool, Camfrog had many Skype features with respect to media and temporal modalities, but was much less sophisticated in terms of security and privacy. This lack of features, however, did not deter users from adopting Camfrog.

SLED users appropriated tools like Camfrog (for SASL) and Skype (for text) into their daily activities more than DCCT users. This could be explained by the fact that the two SLED offices needed to communicate with each other because of geographic distance. Furthermore, end-users at both SLED offices had very similar attributes, e.g. education, PC literacy, PCs with web cams on their desks and broadband connectivity that encouraged the appropriation of a tool like Camfrog. On the other hand, while most DCCT participants had a PC on their desk at offices throughout the Bastion building, face-to-face contact was preferable to and more convenient than virtual contact. More importantly, DCCT-resident end-users had much more ICT4D access than off-site Deaf and hearing users in their potential connectivity circle, especially with regarding to physical access to ICT. Therefore, the connectivity circle at the Bastion was artificial at best. Even though the Deaf telephony prototypes may have functioned adequately, there was no need or even genuine opportunity for DCCT participants to use the tools, as was the case at SLED. The lack of take up of research prototypes at DCCT and the simultaneous take-up of Skype and Camfrog at SLED demonstrated that the social considerations were fundamentally more important than the technical attributes of any ICT4D system we could devise.

5.3 Community based Action Research and Industrial Design approaches (from 2008)



Mobile Gestures Deaf people prefer to use sign language to communicate with each other. There are problems with the video quality when using the real-time video communication available on mobile phones. The alternative is using text based communication on mobile phones however results from other research studies show that Deaf people prefer using sign language to communicate with each other rather than text. This project looked at implementing a gesture based interface for an asynchronous video communication for Deaf people. The gesture interface was implemented on a store and forward video architecture since this preserves the video quality even when there is low bandwidth. The table below provides an overview of the cycle.

Cycle overview	Description
Timeframe	2008–2010
Community	DCCT
Local champion	N/A
Intermediary	Meryl Glaser (SLED), DCCT staff
Prototype	VideoChat
Coded by	Tshifhiwa Ramuhaheli (UCT)
Supervised by	Edwin Blake (UCT)
Technical details	

Mobile gestures cycle overview

Diagnosis

This research builds on the video communication applications developed for Deaf people. Previous systems used a mouse and keyboard to select the desired options. When users are signing they have to sit at a reasonable distance from the camera and the computer so they would have to often move forward to use a mouse or keyboard.

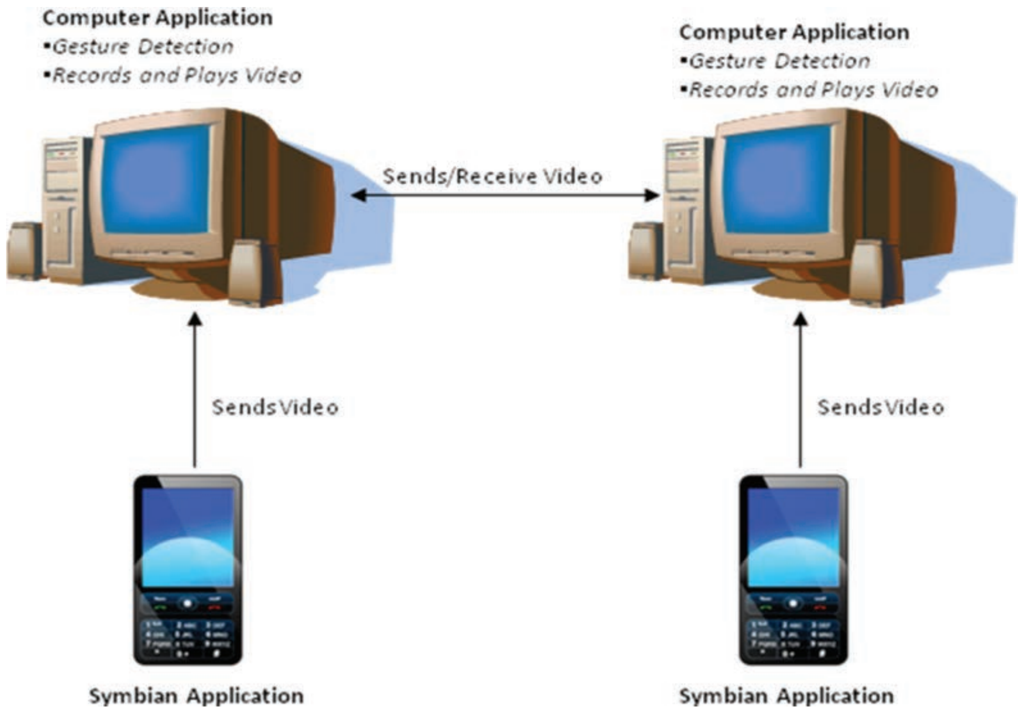
Previously video chat was only on computers and the intention was also to extend the interface to mobile devices. It is clear that the computer of choice in the developing world is the mobile phone. Mainly because of the need to reduce power consumption the processing power of mobile phones has not increased as spectacularly as that of computers.

Plan Action

The objective was to find out if a gesture based interface could improve the usability of an asynchronous or store and forward video communication for Deaf users. We wanted to investigate whether using an interface that can be controlled using hand gestures would make it easier for Deaf people to communicate with each other. By implementing this interface the users are able to control the application from a comfortable signing distance.

The other objective was to extend video communication to mobile phones. However on most mobile phones the camera with better video quality in terms of resolution and frame rate is at the back of the phone. This quality is needed for sign language video communication. Unfortunately this introduces the problem of how to display the video because when the camera faces the user, the phone screen faces away from the user. We investigated whether a television can be used together with the rear camera to record video.

Requirements were gathered in an ongoing fashion throughout the project. The researcher learnt sign language at the accredited Sign Language Education and Development (SLED) centre. The researchers make weekly visits to Deaf community at the Bastion for the Deaf in Newlands. These visits were both about gathering information for the research and assisting the community with ICT needs. Although the researcher was learning sign language a professional interpreter was used during interviews, focus groups and evaluation sessions. During the requirements gathering stage a focus group study was conducted with Deaf users to get feedback on the current video communication. A frequent comment by users was that a touch screen might be more effective than having to use a mouse.



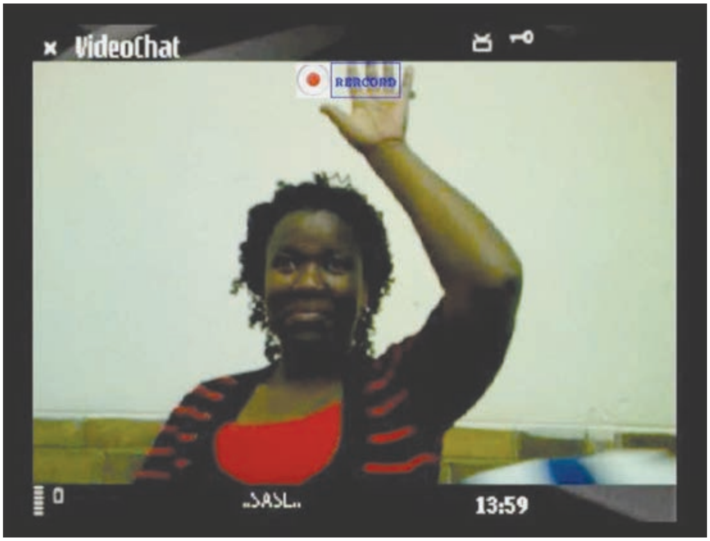
Overview of how the computer and mobile prototype work The computer screen serves in the place of a TV to display the video. The phone is essentially used as a camera. The setup mimics a situation where a powerful phone uses a TV as its video output.

Implement Action

The main iterative stages were:

1. Computer Prototype: try out gesture-based interface
2. First Mobile Phone Prototype: all processing on the phone
3. Second Mobile Phone Prototype: phone off-loaded processing onto PC (to mimic a situation where more powerful phones were available: see above).

The gesture based interface was designed to be similar to a touch screen in the sense that users just have to move their hand to a certain marked area displayed on the screen (see figures below). Instead of touching the screen at the marked area they just have to move their hand in front of the camera. The background of the screen displays a video of the user and the marked areas so when a user moves their hand they can see its corresponding position on the screen at real time. Once their hand is on the desired marked area they have to hold it up for a second until it is detected. This is implemented in order to cater for the situation where a user accidentally put their hand on the marked area while signing. If a hand is moved too quickly the gesture is ignored as it is assumed that the user is signing and they accidentally moved their hand to the marked area.



The user is selecting the record option in order to start recording a video message The gesture based interface was designed to be similar to a touch screen in the sense that users just have to move their hand to a certain marked area displayed on the screen. Instead of touching the screen at the marked area they just have to move their hand in front of the camera.

The user is recording a video message The two options that are available are **cancel** (on the left) which cancels the recording if it is selected and **send** (on the right) which sends the video to the other user.

The user selected send and a confirmation message is shown at the bottom of the screen

The user selected cancel and the confirmation screen is displayed asking the user if they are sure they want to cancel recording the video message

Evaluate Action

1. Computer Prototype: evaluated with Deaf users to determine if the gesture-based interface was useful.
2. Mobile Phone Prototype: not evaluated with the Deaf users because it did not meet the performance requirements required in order to produce a usable interface.

3. Simulated Mobile Phone Prototype: The users had to evaluate if the prototype was usable and effective in facilitating sign language video communication. The prototype was evaluated using a questionnaire, observations and interviews.

Reflection/Diagnosis

The users liked the new way of interacting with the video communication prototype and thought that it made it easier to communicate using video. Although there was a problem with the quality of the video when users were signing fast the users were able to see the signs in the message. The video quality was not what the users would have preferred but they thought that it was good enough to communicate with each other as long as the users were not signing fast.

SignSupport v1
- Mock up of a Deaf communication aid
- Canned video of a mobile device in a PC web browser

Cycle overview	Description
Timeframe	2008–2009
Community	DCCT
Local champion	N/A
Intermediary	Meryl Glaser (SLED), DCCT staff
Prototype	SignSupport
Coded by	Koos Looijesteijn (TU Delft)
Supervised by	Adinda Freudenthal (TU Delft), Henri Christiaans (TU Delft), Edwin Blake (UCT), William Tucker (UWC), Meryl Glaser (SLED)
Technical details	Looijesteijn (2009), Freudenthal and Looijesteijn (2008)

SignSupport v1 cycle overview

Industrial Design approaches were brought into the project to complement the action research by applying context and user analysis methods before starting another design round. We started from the reflections on earlier work to design a telecommunication solution for Deaf-to-Deaf communication. However, as in every industrial design assignment, the design was not started immediately, but first a thorough investigation was conducted. It is important to first step back and check whether the right question is asked, and to understand the user needs and societal context for design. For this the communication problems of the Deaf community were studied in a very general manner: field research about the South African context; a literature review about being Deaf in South Africa; cultural probes (Mattelmäki and Battarbee, 2002) and context mapping (Sleeswijk Visser, 2009) with generative tools (Sanders, 2001); and analysis of data.

We found that there is a need for telecommunication between Deaf people, but that most of the problems they pointed out had to do with communicating with hearing people. For this reason, the design assignment was reformulated as: Design a solution that supports Deaf illiterate South Africans to overcome communication problems with hearing people (Looijesteijn, 2009).

The technical solutions should support various types of communication to hearing people. Finding a solution for talking to just anyone is technically not solvable in a reasonable time frame. Therefore, the ICT solutions will be built up module by module. We focused on the platform and on one example application, talking with a doctor.

Diagnosis

Two diagnosis rounds were conducted in preparation. An approach new to the project was taken: In particular the use of 'generative tools' made a big difference.

Generative tools are drawing/building materials used in a focus session to make a visual representation of a task or a situation under discussion (see figure below). In this investigation Deaf participants were asked to create a visual artefact about day-to-day experiences regarding communication problems, and about communication 'dreams' – how they want it to be. The participants discussed amongst each other the visualized stories from their lives. The research data consists of what the Deaf 'said' to each other. This was translated by an interpreter and recorded on video. Because the researcher could follow the conversations he could occasionally ask questions. The clarifications were also part of the research data.

Deaf participants using generative tools

Sleeswijk Visser (2009) explains that by asking participants to use generative tools and making a visual artefact, a deeper level of knowledge can be uncovered. These deeper knowledge levels are hard to reach by other means, such as interviewing. Tacit knowledge can be revealed (knowledge a person is not aware of about himself), i.e. non-verbal knowledge. Also made explicit are 'obvious elements of life' which are not easily mentioned in interviews because the participant does not realize it might be relevant for the researcher.

A visual artefact produced from using generative tools

Cultural probes were used prior to the generative tools session, in order to introduce participants to the unconventional approach. They were given some creative homework assignments.

Context mapping refers to the designer making a (mental) map about the context of use. This is done by fusing design research data in his/her mind. The data came from the generative sessions (it is not the creative artefacts, but the discussions between the participants that composes the data), from the cultural probes, and also from an ethnography (Fetterman, 1998) to understand the South African context and a literature review about being Deaf in South Africa. This was needed because the designer was Dutch, so he was unfamiliar with the South African context.

A key finding from the session was that the Deaf people want an aid to communicate to each other, but more importantly, most of the problems they pointed out had to do with communicating with hearing people. This became our design goal, viz. a communication aid from Deaf-to-hearing. Other details were uncovered, e.g., the Deaf participants explained that in South Africa doctors always wear a mouth mask, therefore they cannot see the doctor's facial expressions. Doctor-Deaf communication is virtually nonexistent. The Deaf participants are very anxious that they will be treated wrongly because of misunderstandings. Another example was about taxis. In South Africa people share taxi buses which have a set route, but sometimes in consultation with the passengers they deviate. Deaf passengers can miss this conversation and end up in a wrong part of town, leaving them walking home. This is very inconvenient, and can also be dangerous because one should not walk just anywhere in Cape Town. Some very nasty experiences were shared with us, which made it perfectly clear to us what their priorities are and also what the local contextual factors are. Many of the problems we uncovered are Deaf related, but they tend to be very much society specific.

From the first investigation it became clear we had to design a system which would assist in communicating with people from various public services. We decided to start by designing a module for communication with a doctor, because this had been indicated as the most serious problem by the Deaf participants. After this a second more focused investigation was conducted, including interviews with doctors working (or who had worked) in South Africa and a physician in training, and a literature review about South African healthcare, including traditional healers.

Plan action

A design was made to support communication with the police, a pharmacist, in a taxi, etc. Technology investigations and field studies revealed that using an advanced mobile phone (or PDA) was the most suitable solution. Initially, we were considering using a PC in the hospital as a platform, but we found out that many physicians in hospitals don't have a PC at their disposal. It would be better to empower Deaf people – and every Deaf person seems to have a mobile phone.

The final technology choices were dependent on the local situation. Internet is extremely expensive in South Africa and our users are poor, so we decided to use canned video set as packages which is cheaper than streaming video. The dialogues we support follow a tree structure because communication with officials, e.g. doctors, is structured along known paths. It should be noted that these 'known paths' are not so easy to determine, because again there is a lot of tacit knowledge involved in such communication. Therefore, an investigation checking these assumptions about doctor-patient communication was performed before starting to design. A structure for the dialogue tree was designed and has now been implemented (see SignSupport v2 elswhere). Furthermore, a user interface and interaction design for doctor and Deaf users were developed.

User interface of Deaf to doctor communication This figure shows canned video and text. The doctor can work with text and the Deaf user can work with SASL. A dialog tree organizes the conversation so that the required sentence is findable depending on current context of conversation.

The overall design of SignSupport v1 mock-up

Evaluate Action

The usability test: A usability test of the prototype running on a PC with partially working software was conducted. There were no unsolvable problems with the interaction or understanding.

Focus group evaluation—Deaf participants' opinion about the concept:

The Deaf participants liked the design, and appreciated its aims very much. Its essential properties touch on their basic needs, and on human rights, both of which are sometimes at risk. The Deaf participants also liked the user interface solution with the canned video and the dialogue style.

A Deaf user with the SignSupport mock-up running on a PC

Reflection

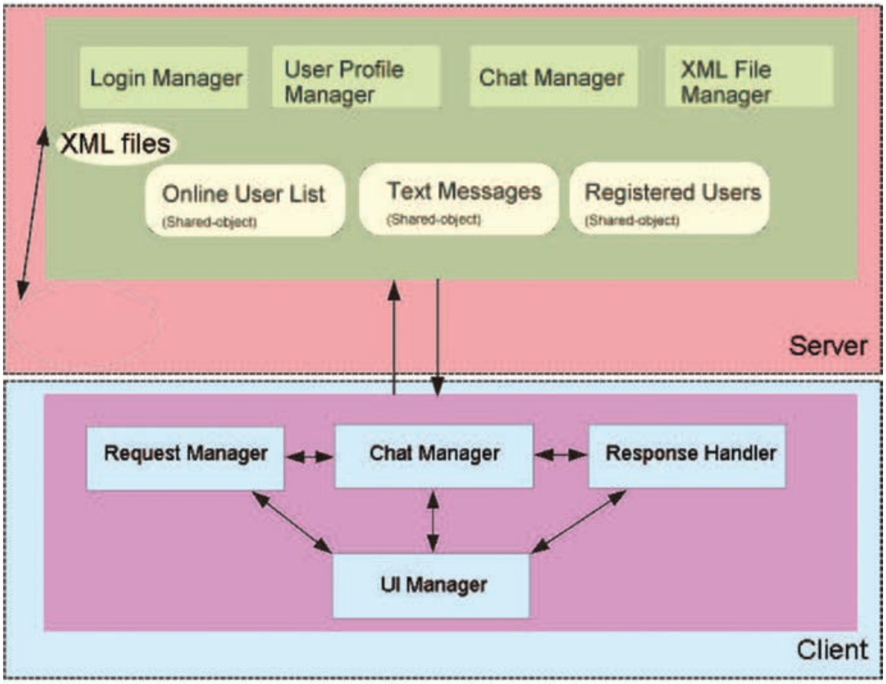
Feasibility assessment: It was estimated that the effort required to bring the design to implementation was too much for a realistic time frame, i.e., because of all the possible dialogue trees. Therefore, we decided that as a next step we would continue with, a somewhat simpler dialogue tree for a pharmacy scenario. This investigation has started. This application will require some different user interface properties. We plan to implement it as the first of a set of applications, ranging from communication with the police, civil services, and doctors. We hope that the South African government will understand the need for such technology and will also recognize the low costs of ICT as compared with human interpreters. Our future prototype is not only meant for user testing but will also be used shown to government. Once its value is established we hope other applications will be sponsored from the government budget for Universal Access.

Deaf Video Chat v2
- Browser-based video
- Mobile phone video

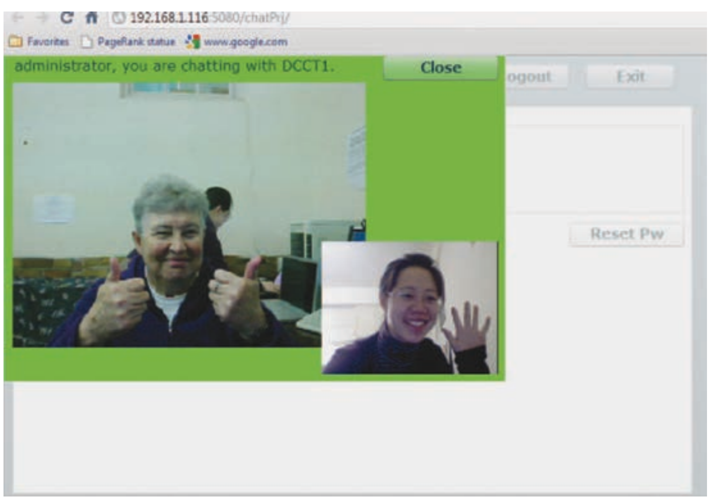
This project offers some prototypes to provide browser-based and mobile video communication services for Deaf people and evaluates these prototypes. The aim of this research is to identify an acceptable video communication technology for Deaf people by designing and evaluating several prototypes. The goal is to find one that Deaf people would like to use in their day-to-day life. The project focuses on two technologies—browser-based systems and mobile applications. Several challenges emerged, for example, specific Deaf user requirements are difficult to obtain, the technical details must be hidden from end users, and evaluation of prototypes includes both technical and social aspects. This project describes work to provide South African Sign Language communication for Deaf users in a disadvantaged Deaf community in Cape Town. We posit an experimental design to evaluate browser-based and mobile technologies in order to learn what constitutes acceptable video communication for Deaf users. Two browser-based prototypes and two mobile prototypes were built to this effect. Both qualitative data and quantitative data are collected with user tests to evaluate the prototypes. The video quality of Android satisfies Deaf people, and the portable asynchronous communication is convenient for Deaf users. The server performance is low on bandwidth, and will therefore cost less than other alternatives, although Deaf people feel the handset is costly. The table below provides an overview of the cycle.

Cycle overview	Description
Timeframe	2009–2010
Community	DCCT
Local champion	N/A
Intermediary	Meryl Glaser (SLED), DCCT staff
Prototype	Prototype-Flash, Prototype-HTML5, Prototype-J2ME, Prototype-Android
Coded by	Yuan Yuan Wang (UWC)
Supervised by	William Tucker (UWC)
Technical details	Wang and Tucker (2009), Wang and Tucker (2010), Wang (2011)

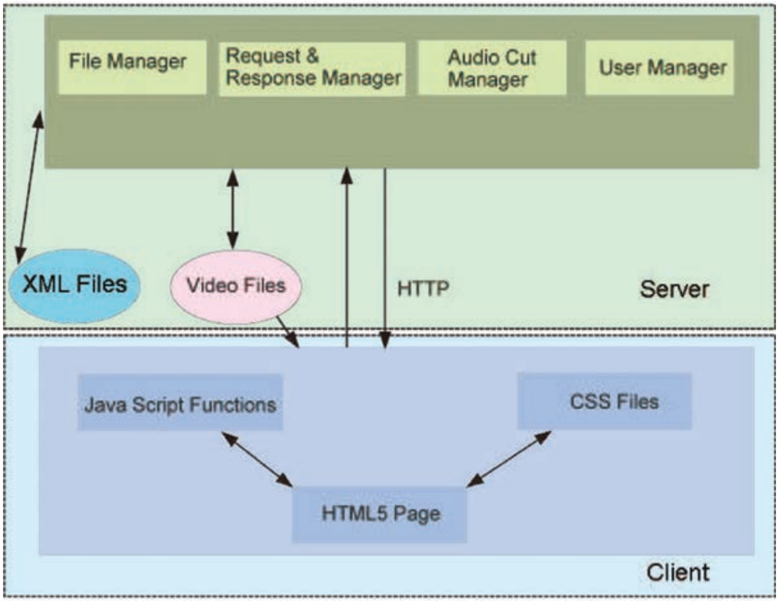
Deaf Video Chat v2 cycle overview



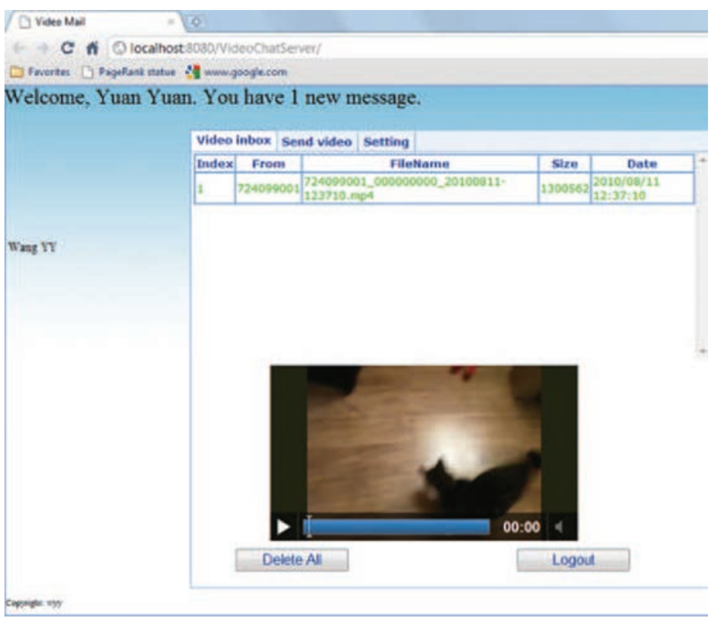
Prototype-Flash architecture



Prototype-Flash user interface

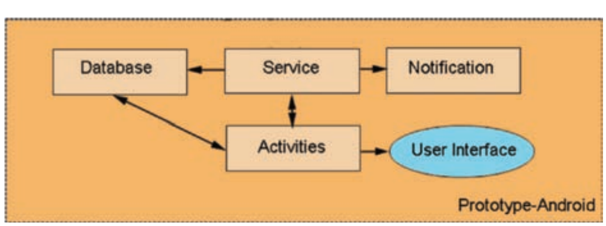


Prototype-HTML5 architecture

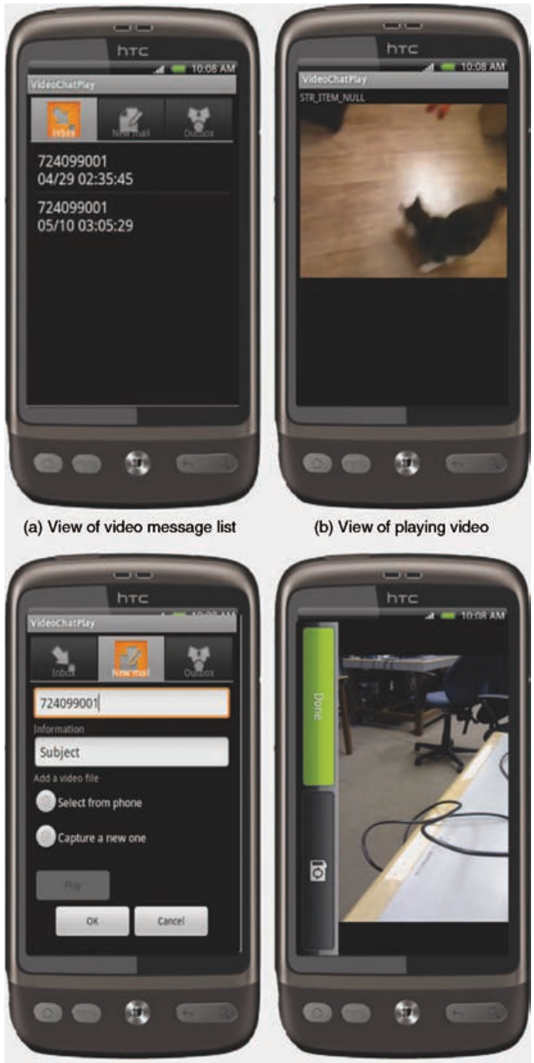


Prototype-HTML5 user interface

Prototype-J2ME user interface



Prototype-Android architecture



(a) View of video message list (b) View of playing video

(c) View of sending video (d) View of mail history

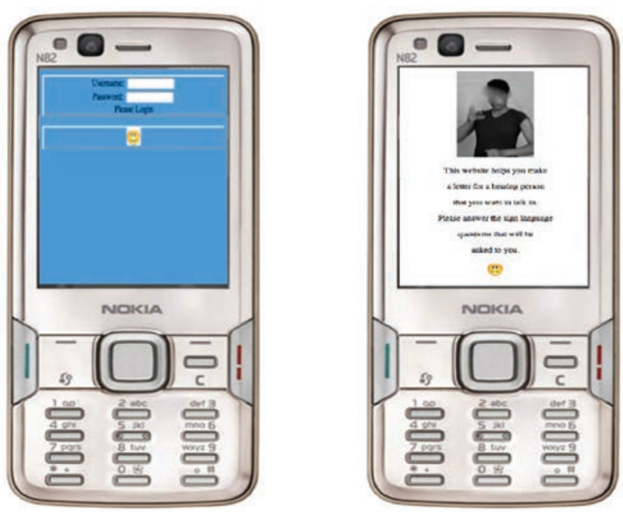
Prototype-Android user interface

SignSupport v2
- Implementation on mobile device
- Generalized architecture to handle additional scenarios

Many Deaf people use their mobile phones for communication with SMSs yet they would prefer to converse in South African Sign Language. Deaf people with a capital D are different from deaf or hard of hearing as they primarily use sign language to communicate. This study explores how to implement a Deaf-to-hearing communication aid on a mobile phone to support a Deaf persons visit to a medical doctor. The aim is to help a Deaf person use sign language to tell a hearing doctor in English about medical problems using a cell phone. A preliminary trial of a computer-based mock-up indicated that Deaf users would like to see the prototype on a cell phone. A prototype will be built for a mobile phone browser using sign language video arranged in an organized way to identify a medical problem. That problem is then identified in English and shown to the doctor with the phone. User trials data will be collected with questionnaires, semi- structured interviews and video recordings. The technical goal is to implement the prototype on a mobile device in a context free manner, allowing the plug and play of more communication scenarios, such as visits to a doctor's office, the Department of Home Affairs or the police station. The table below provides an overview of the cycle.

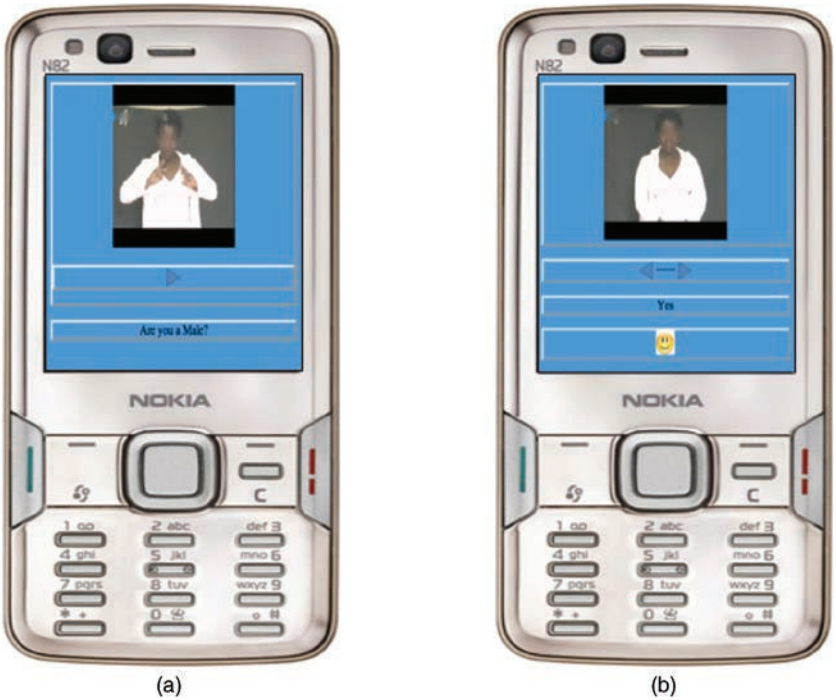
Cycle overview	Description
Timeframe	2009–2010
Community	DCCT
Local champion	N/A
Intermediary	Meryl Glaser (SLED), DCCT staff
Prototype	SignSupport
Coded by	Muyowa Mutemwa (UWC)
Supervised by	William Tucker (UWC)
Technical details	Mutemwa and Tucker (2010)

SignSupport v2 cycle overview

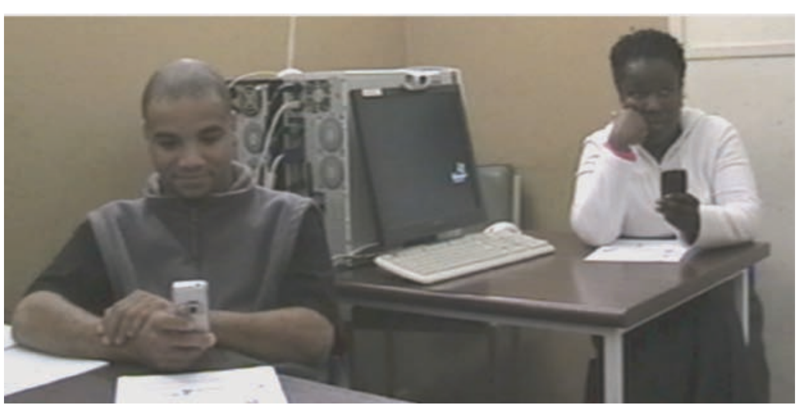


(a) (b)

Introduction and login screens The Deaf user's login screen (a), after entering in the username and password the Deaf user continues to the next page by clicking on the smiling image. The introduction page (b) displays a SASL video and its English equivalent which describes to the Deaf user what the system is all about and how s/he can use it.



Question screen with video embedded in XHTML and an answer screen The question screen (a) has video embedded into an XHTML page playing inside a mobile browser using Adobe Flash player. The English text equivalent appears below the video and the navigation arrow is between the SASL video and the English text. A page displaying an answer (b) has a SASL video and its English equivalent describing the answer. The Deaf user can navigate to the previous page using the left arrow and to the next page using the right arrow, or can accept the answer by clicking on the smiley face.



Two Deaf users testing SignSupport

New scenario creation screen To create a new scenario, the user must click an option. Option 1 creates the introduction page. Option 2 adds questions and their answers. Option 3 generates the hearing persons screen. Option 4 creates the response pages. Option 5 creates the English summary pages.

6. References

Blake EH and Tucker WD (2004). Bridging communications across the digital divide. *Proc. 3rd Workshop on Social Intelligence Design,* (*SID 2004*), Enschede, Netherlands, 29-38.

Campbell B, Rosenberg J, Schulzrinne H, Huitema C and Gurle D (2002). Session Initiation Protocol (SIP) extension for instant messaging. RFC 3428.

DoC (1996). Telecommunications Act No. 103 of 1996. President's Office, Republic of South Africa, No. 1869.

DoC (2001). Telecommunications Amendment Bill 65 of 2001. Minister of Communications, Republic of South Africa, No. 22630.

DoC (2005). Draft Convergence Bill B9 of 2005. Government Gazette, Republic of South Africa, No. 27294.

Fetterman DM (1998). *Ethnography, Step by step*, second edition, Applied Social Research Methods Series Vol 17, Sage Publications, Inc, Thousand Oaks, CA.

Freudenthal A and Looijesteijn K (2008). Co-designing for special user groups in developing countries: the Deaf in South Africa. Position paper for: Building an International Community: Designing Interactive Systems (DIS) for the Developing World workshop, 22-24 Feb 2008, Cape Town, South Africa.

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, (CD-ROM publication).

Glaser M and Tucker WD (2001). Web-based telephony bridges for the Deaf. *Proc. South African Telecommunications Networks & Applications Conference,* (*SATNAC 2001*), Wild Coast Sun, South Africa, 672-677.

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, (CD-ROM publication).

Glaser M, Tucker WD and Mashao D (2004). Preparation of Deaf end-users and the Softbridge for semi-automated relay trials. *Proc. South African Telecommunications Networks & Applications Conference,* (*SATNAC 2004*), Stellenbosch, South Africa, II-255-256.

Glaser M, Young P and Porteous L (2005). An information and communication technology (ICT) needs analysis and training programme for Deaf people. *Proc. 20th International Congress on the Education of the Deaf,* (*iced 2005*), Maastricht, Netherlands, 231.

Handley M, Schulzrinne H, Schooler E and Rosenberg J (1999). SIP: Session initiation protocol. RFC 2543.

Hersh MA and Tucker WD (2005). Ethics and mono-disciplinarity: Positivism, informed consent and informed participation. *Proc. 16th International Federation of Automatic Control World Congress,* (*IFAC*), Prague, Czech Republic.

Julius EP (2006). Guaranteed delivery of multi-modal semi-synchronous IP-based communication. Unpublished MSc, University of the Western Cape, Bellville, South Africa.

Julius EP and Tucker WD (2005). Guaranteed delivery of semi-synchronous IP-based communication. *Proc. South African Telecommunications Networks & Applications Conference,* (*SATNAC 2005*), Drakensburg, South Africa, I-407-412.

Lewis J, Tucker WD and Blake EH (2002). Softbridge: An architecture for building IP-based bridges over the digital divide. *Proc. South African Telecommunications Networks & Applications Conference,* (*SATNAC 2002*), Drakensburg, South Africa, 383-386.

Lewis J, Tucker WD and Blake EH (2003). Softbridge: A multimodal instant messaging bridging system. *Proc. South African Telecommunications Networks & Applications Conference,* (*SATNAC 2003*), George, South Africa, I-235-240.

Looijesteijn K (2009). The design of a deaf-to-hearing communication aid for South Africans, MSc thesis, TU Delft.

Ma ZY and Tucker WD (2007). Asynchronous video telephony for the Deaf. *Proc. South African Telecommunications Networks and Applications Conference,* (*SATNAC 2007*), Mauritius, 134-139.

Ma ZY and Tucker WD (2008). Adapting x264 to asynchronous video telephony for the Deaf. *Proc. South African Telecommunications Networks and Applications Conference,* (*SATNAC 2008*), Wild Coast Sun.

Mattelmäki T and Battarbee K (2002). Empathy probes, *Proceedings of PDC2002*, Malmö, June, 23-25.

Mutemwa M and Tucker WD (2010). A mobile Deaf-to-hearing communication aid for medical diagnosis. *Proc. South African Telecommunications Networks & Applications Conference,* (*SATNAC 2010*), Stellenbosch, South Africa, 379-384.

Office of the Deputy President (1997) White Paper on an Integrated National Disability Strategy. Western Cape: Rustica Press.

Penton J (2003). An empirical, in-depth investigation into service creation in H.323 version 4 networks. Unpublished MSc, Rhodes University, Grahamstown, South Africa.

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.

Sanders EBN (2001). Harnessing People's creativity: Ideation and Expression through Visual Communication, in Focus Groups: Supporting Effective Product Development Langford J and McDonagh-Philip D (eds.),Taylor and Francis.

Sleeswijk Visser F (2009). Bringing the everyday life of people into design. Doctoral thesis, TU Delft.

Sun T (2005). Carrier-grade adaptation for an IP-Based multimodal application server: Moving the Softbridge into SLEE. Unpublished MSc, University of the Western Cape, Bellville, South Africa.

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.

Tucker WD (2003). Social amelioration of bridged communication delay. *8th European Conference of Computer-supported Cooperative Work,* (*ECSCW 2003 Doctoral Colloquium*), Helsinki, Finland.

Tucker WD (2004). Connecting bridges across the digital divide. *Proc Human Factors in Computing Systems,* (*CHI '04*), Vienna, Austria, 1039-1040.

Tucker WD, Glaser M and Penton J (2002). A bridge for the problem of Deaf telephony. *Science in Africa*, June 2002, http://www.scienceinafrica.co.za/2002/june/deaf.htm.

Tucker WD, Blake EH and Glaser M (2003a). Building bridges for Deaf telephony in South Africa: A community-centred approach. *IFIP Working Group 9.4 Newsletter*, 13(2), http://www.iimahd.ernet.in/egov/ifip/nov2003/nov2003.htm.

Tucker WD, Blake EH and Marsden G (2004). Open user interconnect and quality of communication. *Proc. South African Telecommunications Networks & Applications Conference,* (*SATNAC 2004*), Stellenbosch, South Africa, II-261-262.

Tucker WD, Glaser M and Lewis J (2003b). Softbridge in action: The first Deaf telephony pilot. *Proc. South African Telecommunications Networks & Applications Conference,* (*SATNAC 2003*), George, South Africa, II-293-294.

Wang YY and Tucker WD (2009). Browser-based video communication for Deaf people. *Proc. South African Telecommunications Networks and Applications Conference,* (*SATNAC 2009*), Swaziland, 489-490.

Wang YY and Tucker WD (2010). Browser-based sign language communication. *Proc. South African Telecommunications Networks & Applications Conference,* (*SATNAC 2010*), Stellenbosch, South Africa, 71-76.

Wang YY (2011). Browser-based and mobile video communication alternatives for Deaf people. MSc thesis, University of the Western Cape, Bellville, South Africa.

Zulu TAZ and Le Roux KF (2005). A preliminary investigation into Text-To-Speech (TTS) software in a semi-automated relay service for Deaf telephony. Unpublished Honours, University of Cape Town, Cape Town, South Africa.