Usability of an Authoring Tool for Generalised Scenario Creation for SignSupport

Lindokuhle S. Duma[1], Prangnat Chininthorn[2], Meryl Glaser[3] and William D. Tucker[4]

[1,2,3,4]Department of Computer Science, University of the Western Cape

Robert Sobukwe Road, Bellville 7535 South Africa

Tel: +27 21 959 3010, Fax: +27 21 959 1274

[1]3063344@myuwc.ac.za
[2]pchininthorn@uwc.ac.za
[3]merylglaser@gmail.com
[4]btucker@uwc.ac.za

***Abstract*— This paper presents the usability testing results for an authoring tool that generalises scenario creation for a tool called SignSupport. SignSupport is a mobile communication tool for Deaf people that currently runs on an Android smartphone. The authoring tool is computer-based software that helps a domain expert, with little or no programming skills, design and populate a limited domain conversation scenario between a Deaf person and a hearing person, e.g., when a Deaf patient collects medication at a hospital pharmacy or when a Deaf learner is taking a computer literacy course. SignSupport provides instructions to the Deaf person in signed language videos on a mobile device. The authoring tool enables the creation and population of such scenarios on a computer for subsequent 'playback' on a mobile device. The output of this authoring tool is an XML script, alongside a repository of media files that can be used to render the SignSupport mobile app on any platform. Our concern now is to iteratively develop the user interface for the authoring tool, focusing on the domain experts who create the overall flow and content for a given scenario. The current authoring tool was evaluated for usability; for both pharmacy and ICDL course scenarios with purposive sampling. The findings suggest that the authoring tool can generalise SignSupport for multiple limited domain scenarios, mobile platforms and signed languages.**

Keywords— **Mobile Apps, Software Design, XML**

I. Introduction

The authoring tool for generalised scenario creation described in the paper is a computer-based application that helps a domain expert with little or no programming skills to design a front-end interface for a communication flow between a Deaf and a hearing person. The end result is a mobile communication tool called SignSupport that helps a Deaf person with a given communication scenario; with pertinent information relayed to the Deaf user in signed language [1]. The authoring tool produces an Extensible Markup Language (XML) script as an output after the flow of a given scenario is designed by a domain expert. This XML file is consumed by an XML parser to render a mobile application on any given mobile platform [1]. The current version of SignSupport is a mobile application that helps a pharmacist to give comprehensible medical instructions to a Deaf patient during medicine dispensing in the form of pre-recorded South African Sign Language (SASL) videos [2]. These medical instructions are stored on a mobile phone's memory card for the patient to view at any time. The limitations of the SignSupport mobile app are that it only caters for the pharmacy setting scenario, and runs only on an Android mobile platform. Therefore, this authoring tool is aimed at generalising SignSupport to accommodate multiple domain scenarios; for multiple mobile platforms and that can be populated by any language for low literacy end users.

This paper uses Deaf with a capital 'D' to refer to a linguistic and cultural group of people with hearing loss who mainly use SASL as a mother tongue [3]. Deaf people often experience communication barriers while communicating with a hearing person who cannot sign. While many technologies, especially mobile, support voice and text communication, this presents usability difficulties for Deaf people with low functional text literacy. SASL interpreters are very expensive for Deaf people as they are very scarce, and Deaf people battle significantly to communicate with hearing people in the absence of an interpreter. SignSupport is an assistive technology meant to bridge such gaps for a given scenario, and the authoring tool is meant to support scenario designers.

The authoring tool was evaluated for usability with participants who were recruited through an applied purposive sampling method. This is because the scenarios were defined prior to usability testing. Participants who were specialised in these domains or have worked with Deaf people before were recruited. The results show that the authoring tool is capable of creating multiple limited domain scenarios and that it supports a domain expert with little or no programing skills.

The rest of the paper is organised as follows. Section II covers the history of SignSupport. Section III covers the related work. Section IV describes the authoring tool prototype. Section V defines the research methods that guide the research and the prototype implementation. Sections VI presents and discusses the results obtained from the prototype usability testing. Section VII concludes and outlines the next research steps and future work.

II. History of SignSupport

Since 2009, there have been several versions of SignSupport produced through an iterative research process.

1) Version 1 was an internet browser-based mock-up design [3]. The main objective of this mock-up was to help the doctor understand the symptoms of a Deaf patient so that s/he could prescribe medication for the patient. The mock-up presented a set of questions in SASL which the Deaf patient

answered before seeing the doctor. The doctor then viewed the summary of the patient's answers in English text.

2) Version 2 was for the same scenario as version one and was developed as a Symbian mobile app with XHTML [4]. This mock-up required only a mobile phone with a data connection running a browser that supported Small web format. However, the doctor and Deaf patient scenario was found to be too wide and complicated to be constructed as an application [5].

3) Version 3 was then shifted into a more constrained communication domain between a pharmacist and a Deaf patient [5]. Its aim was to help the Deaf patient understand the medical instructions as prescribed by a pharmacist. This mock-up was designed with two types of interfaces: one for the Deaf patient to input background information prompted by SASL videos and icons and one for the pharmacist to view the Deaf patient's background information and dispense medication. The Deaf interface also relayed that information in pre-recorded SASL videos.

4) Version 4 was a redesign by a multi-disciplinary and trans-university team and then implemented as an Android app [6]. The construction of this version is now in the process of usability testing in a public hospital pharmacy.

The SignSupport approach was also used to create a prototype to aid with self-paced learning for an International Computer Driving License (ICDL) course. That prototype was developed in parallel with the more recent pharmacy prototypes. Our research team sees the potential of SignSupport for additional contexts, i.e. other limited domain communication scenarios. Therefore, this paper proposes that an authoring tool for SignSupport can provide a solution to assist domain experts to design a communication flow that can meet specific needs for Deaf end users.

III. Literature Review

This section discusses the literature review that has been consulted in order to formulate this study. We used the ideas and techniques that that were found in this literature review, as references when building and evaluating the authoring tool. This ensured that the authoring tool suits the needs of the domain experts and that it provides the greatest user experience when in use.

A. Authoring tools in general

An authoring tool allows a user with little or no programming experience to design and implement complex applications, mainly web applications [7]. Authors can create these complex and attractive applications by merely clicking and defining relationships between objects, e.g., text, pictures and videos. An authoring tool also allows an author to preview a design to see how it will look when it is a complete app. Examples of well known authoring tools include Dreamweaver and Microsoft Front Page. The structure and the features of the existing authoring tools were used as a reference to inform the interface design of the SignSupport authoring tool. One typical similarity amongst authoring tools is that they have multiple distinct templates that authors can customise directly or by changing the templates' source code. Templates help authors to work faster as they are generic and can be adapted to any form within the intended use of that authoring tool.

B. Language independence

Language independent techniques involve developing applications that accommodate different human languages. This means that software assets (text, icons and videos) can be easily changed or replaced. Language independent software accommodates different data formats and conforms to software engineering principles such as software reuse [8]. The SignSupport authoring tool will have screen templates that can be modified and reused limitlessly with different data assets. Pictures and videos have different data types so this authoring tool will use generic libraries that can read most popular data formats.

C. Design for all

This method is also known as universal design, as it entails creating software that can be used by all people effectively, without the need for adaptation or specialised resources [9]. Principles of this method include creating useful designs that accommodate a wide range of individual preferences and abilities, error tolerance and low physical effort [9]. The aim is also to accommodate Deaf domain experts, with low text and functional literacy, as authoring tool users in the future. The techniques that are used to achieve this goal include direct manipulation via drag-and-drop of objects in a WYSIWYG editor [10]. The authoring tool has tooltips, help options and a user guide for novice users. This enables any author to learn to use the authoring tool quickly and also to have assistance within the authoring tool in case s/he needs it. These helper options can also be adapted to in signed language videos, as tutorials that demonstrate certain functions for Deaf authors.

IV. The Authoring Tool for SignSupport

This section describes the features available in the authoring tool. The authoring tool helps a domain expert to design an interface for a limited domain conversation flow between a Deaf and a hearing person. The software presents the domain expert with screen templates. These screen templates can be populated with text, icons and/or signed language videos (that would be provided by native signed language speakers, not the domain experts, unless they are Deaf, of course). The screens can also be linked to each other in a graph structure. The link between each screen indicates the user interaction on the real mobile app i.e. which screen will show next when the end user clicks a given button. Figure 1 presents a screenshot of the authoring tool interface while a domain expert is using it. The interface of the authoring tool is divided into four areas:

1) The button area contains four buttons, namely 'Add template', 'Clear all', 'Add video' and 'Export to XML'. When the domain expert clicks on the 'Add template' button, a window pops up and presents all the default screen templates of the authoring tool. The user then chooses the template that s/he would like to add to the canvas by clicking the checkbox on the left side of the preferred template and then clicks the 'OK' button at the bottom of the window. The selected screen template then appears on the canvas as active. The 'Clear all' button clears all the screen templates from the canvas. The 'Add video' button helps the domain expert add videos from the computer as an asset within the authoring

tool. The videos line up vertically as they are being added from the computer to the video area. The 'Export to XML' button invokes a method that reads all the screen templates, their assets and connections to each other; and then produces an XML file that includes all of this information. Since the XML file contains only text, it only stores an asset's information such as filename and relative path, in text. An XML parser on a target device consumes the XML file in order to render a platform-specific mobile app.

2) The video area contains a series of sign language videos that are added by the domain expert from the computer. The sign language videos can be recorded before or after the scenario based conversation flow is designed. Then each video is given an English tag/filename in a way to help the scenario domain expert include the correct video in the right place [11]. Videos in this area can be played (and paused) by clicking on them. They can also be dragged and dropped to other screens.

3) The icons tab area contains icons in various categories. The domain expert can also add new tabs of icons. The icons can be dragged and dropped to the screens, and linked to other screens and/or actions.

4) The Canvas area is where the selected screen templates appear when added. The domain expert populates a screen by typing in the intended message and dragging and dropping icons or a video to the indicated areas in the screen. If the domain expert drops an icon into a wrong area, s/he can right click on that area and select the 'remove icon' option and the icon will be removed. To connect the screen to another, the domain expert clicks on the icon that represents a button from one screen, confirms that s/he wants to connect that icon to another screen at a prompt dialog pop-up, then selects the screen to connect with. A blue link line will appear in the Canvas, between these two screens. This link gets updated and redrawn as the screens are moved from one position to another.

The authoring tool is built using the Java programming language, and the output of the authoring tool is an XML file. The XML file serves as a database that stores the design information in text. XML is supported by a variety of technologies in various platforms [12]. This in turn helps us to achieve platform independence for scenarios designed from the authoring tool.

V. METHODS

This research follows an action research methodology [13]. This approach is cyclic in nature and involves an iterative series of problem definition, planning, implementation, observation and reflection phases. The aim of action research is to improve or change both technological and social systems for the better [13]. We hope the SignSupport mobile app proves to be a useful communication tool for both Deaf people and pharmacists. Deaf people also feel that they need communication tools for other scenarios which they can install on their various mobile phones, hence the need for an authoring tool. The development of the authoring tool is based on an iterative prototyping and usability testing method. The first step is to iteratively implement core features of the authoring tool and then evaluate them through usability testing before adding more features. Testing our prototype for usability helps us to confirm user requirements, uncover software bugs and to accommodate new features that end users may suggest after trying out the prototype. The domain experts are the main drivers of the authoring tool implementation. Since the whole research is guided by action research methodology, it is important that they are included throughout the research process.

User-based testing was employed for the usability experiment [14]. The goal of the experiment was to evaluate how effective the authoring tool can be, in enabling a domain

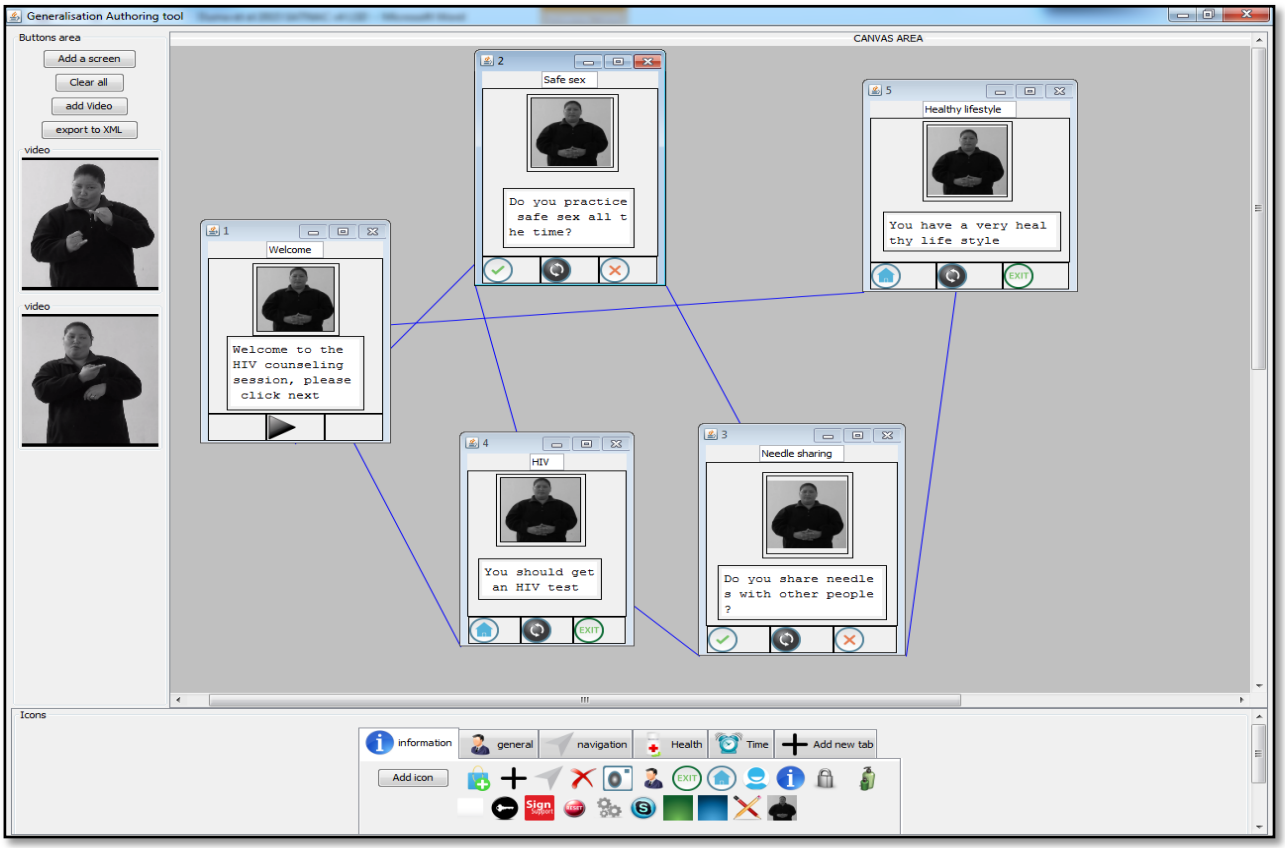

Figure 1: A screen shot of the Authoring tool's interface while a 'user' (domain expert participant) is testing it.

expert with little or no programming skills, to create a limited domain conversation scenario. The main focus is how the domain expert searches for icons from the icon tabs, adds and removes icons and videos from the screen templates, recovers from errors; reuses screen templates in the same scenario and reuses the same icons in different screen templates are also observed. The domain expert was also be encouraged to give suggestions on how the prototype can be improved to better suit their needs.

A. *Sampling*

Participants were selected through a purposive sampling method. Purposive sampling is purposefully selecting participants in terms of the qualities/skills they have [15]. The communication flow design scenarios for the experiment had already been defined, as for the medication dispensing process and the (ICDL) course. The pilot session of the authoring tool was evaluated with the aid of an industrial design engineer. Participants recruited were a pharmacist for the pharmacy setting scenario, a Deaf educator and a computer science researcher for the ICDL course scenario for the actual usability testing sessions. All participants have unique work experience with Deaf people. The industrial design engineer designed the SignSupport mobile app and the pharmacist is involved in the co-design and co-testing of the previous and current versions of the SignSupport mobile app with Deaf users. The Deaf educator gives ICDL lessons in SASL every Thursday at a local Deaf community; and the Computer science researcher works on another authoring tool for creating ICDL lessons only. All our participants are the potential users of this authoring tool and hence ideal candidates for the experiment.

B. *Testing procedure*

The testing was divided into three stages, namely training, testing and gaining user feedback. One participant attended the usability test at a time. During the training stage, the participant was introduced to the authoring tool prototype and all the features were demonstrated. The researcher then encouraged the participant to try out the features on their own e.g. add screens to the canvas area, drag and drop icons and videos to the indicated area, and link icon labels with other screens. Afterwards a training exercise was given to the participant to design an HIV counselling scenario. This applied to all participants. This scenario was given to the participants as a decision flow graph diagram, as shown in Fig 2.

During the testing stage, participants were first asked to design and populate a limited scenario that was specific to their domain. The participants from the pharmacy scenario were asked to design a few screens from the familiar SignSupport mobile app for medicine dispensing. The participants from the ICDL course scenario were asked to design a specific lesson from the ICDL course book, which can be populated with SASL video instructions. The participants were assigned the tasks specific to their domain to give them a feeling of how they would use the authoring tool to best suit their needs. Each participant was asked to speak out about everything they see, think, and act while using the authoring tool. All the 'think-out-loud' [13] messages were accounted as part of their feedback. This process was then followed for the HIV scenario for all of the participants.

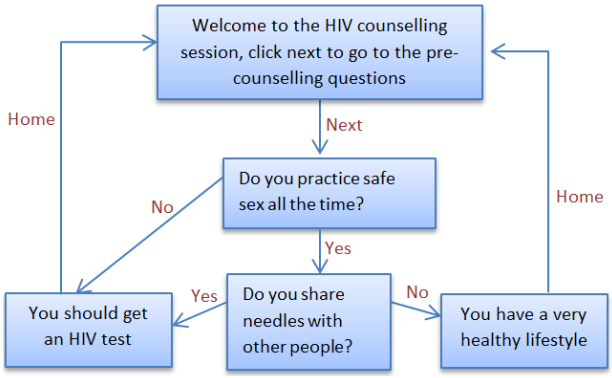

Figure 2. HIV pre-counselling scenario used for the training stage for all participants

During the feedback stage, additional questions regarding the available features and the use of the authoring tool were asked to the individual participants for additional clarification.

C. *Data collection*

Qualitative methods were used to collect data. The usability session was video recorded. The video recorder was directed to the computer screen to record how the participant interacted with the authoring tool. Participants were asked to use the think aloud strategy to voice out what they were doing and all the thoughts that came to their mind as a result of using the tool. After the testing stage, the researcher conducted a semi structured interview with each participant. The main questions asked in the interview were:

- What did you like/dislike about the authoring tool?
- What did you find easy to use when building the scenarios?
- What did you find confusing or difficult?
- Any features you would like to be changed or added to the authoring tool?

D. *Data analysis*

Data collected from the trial was summarised and organised according to the four areas of the authoring tool interface, which are stated and discussed in the section below.

VI. Results and Discussion

All participants finished the training scenario successfully and with no difficulties. They all repeatedly used one screen template with the same icons for the training scenario design. All of them also managed to create and populate the testing scenario design. The participants' feedback from both the pilot and the actual usability test session is described in accordance with the aforementioned four areas of the authoring tool interface features.

A. *Buttons area*

Here is the description of feedback from all the participants regarding the available buttons in the button area. All participants understood the use of each button on the interface.

1) 'Add screen': All participants praised the interface and notification designed for the 'add screen' and its process. The selecting process was easy to understand, the background colour of the selected screen template changed as it was clicked, and this assured the participants that the template was indeed selected.

2) The canvas was cleared immediately after clicking the "Clear all" button; and the XML output was created and saved to a file after clicking the "Export to XML" button. Two participants were also interested in seeing what the XML output looked like.

3) Suggestion on the additional button: Two participants said that there should be an 'Add picture' button that they can use to add pictures to the screens, which works similarly to the 'Add video' for adding videos to the screen.

B. Video Area

This section describes feedback obtained about the video area.

1) Adding video to the video area: All participants found the video addition process to the video area easy to do.

2) Suggestion of a scrollbar: One participant suggested that there should be a scrollbar in the video area, so that multiple videos which will be added to the video area can be seen.

3) Suggestion of video controls: One participant suggested that there should be 'video control buttons' that show on each video in the video area, when the mouse is hovered over them, so that he could click on these video controls to play each video.

C. Icon tabs area

All participants could navigate through the icon tabs easily. Names of all the icons appeared when the participants moved the mouse over them. Two participants commented that the tooltip assured them that they were choosing the icon with the correct semantic meaning to drag to the screens. Two participants struggled to find the 'home' and the 'exit' icons as they expected to find these icons under the navigation icon tab instead.

D. Canvas Area

This section presents the feedback regarding the screen's components when the screens were already added to the canvas area.

1) Adjustment of the screen and its component sizes: All participants disliked that the components of the screen remained in the same size and position when the screen was resized. They then had to resize and position each of them one by one. So, two participants suggested that the whole screen and its components should be proportionally resized all at once.

2) The design of the default screen templates: One participant found the default screen templates limited in terms of design. Most of the component areas were fixed in their position, which did not allow the user to modify the default design. One participant mentioned placing the 'next' and 'back' buttons on the sides of the screen, whereby the default screen template did not match this need. Therefore she suggested having a blank screen template available for users to customize to their own screen design(s).

3) Screen links: All participants found the screen linking option easy to understand and use; however, they preferred that the linking process be done with fewer mouse clicks. The linking line should go around the screens which lie between the two indicated screens instead of going underneath them. With the suggested linking line, it would be easier for the user to trace back and recheck each link. In addition, one participant suggested that the line indicating the link should have a message that indicates its origin and the destination of linking.

4) Drag and drop option: All participants said that they liked the drag and drop options for adding icons and videos to the screen. It was observed that all participants missed the icon drop area several times when they were populating the screens with icons. However they tried again even though the authoring tool did not give them any warning/notification

The table below lists features that the participants suggested the authoring tool should have. Participants made these suggestions during the feedback session, when they were asked about what they would add to the authoring tool. These features will be added to the authoring tool and will also be evaluated for usability in the next iteration.

TABLE 1

SUGGESTIONS OF ADDITIONAL FEATURES OBTAINED FROM THE PARTICIPANTS

Feature(s)	**Motivation**
Zoom in and out	Helps authors to their whole scenario design outline without having to scroll up or down. This feature will also enable the author to focus on one screen or part of the screen at a time
Copy and paste	To duplicate screens that are already populated; and modify them for a similar purpose so that there won't be a need to start over all the time.
User guide and help function	To enable the author to learn to use the authoring tool independently and to get assistance in the absence of the researcher.
Tooltip text on buttons and mouse-hovers	Tooltip text on a button or mouse-hover option reminds the author what each button or option is used for.
Scenario design simulation	This gives authors an idea of how the scenario would look like as a real app and also to confirm that the links have been defined correctly.
Undo and redo	This option helps to recover from errors or to repeat recent events.

Despite all the recommendations for improvement, all participants felt that the authoring tool could assist and empower their capacity greatly to create a communication flow to use with a Deaf counterpart. All participants praised the authoring tool, saying that it was easy to use and it saved

a lot of time since most of the usability session tasks were completed within few minutes, by just clicking and dragging items to the screens in the canvas. They also mentioned that they would like to use it in the future.

VII. Conclusion and Future Work

The results indicate that there are several additional features required from the authoring tool to improve its usability. The robustness of the authoring tool needs to be thoroughly evaluated before the taking it to its potential users for usability testing of the next iteration. The creation of the three limited domain scenarios (HIV pre-counselling, ICDL course lessons and rebuilding a thread of the SignSupport pharmacy app) demonstrates that the authoring tool can generalise SignSupport scenario creation for limited domain communication between Deaf and hearing people. The reuse of screen templates and data assets (icons and videos), combined with the production of an XML file representing a given scenario, appear to indicate that the authoring tool can generalise the SignSupport mobile app for multiple signed languages and mobile platforms.

The next step is to improve the authoring tool by fixing its errors and adding features recommended by the participants. Another usability testing exercise will then be conducted with more participants and more limited domain scenarios. Thereafter, one complete limited domain scenario will be built using the authoring tool, with recorded SASL videos with the help of a Deaf person and a SASL interpreter. Then that scenario can be evaluated with Deaf participants. That requires a fully functioning XML parser. The XML parser will enable the authoring tool to produce a mobile app that can be run on a mobile phone. Note that signed language videos will have to be recorded separately, then integrated into the scenario. Another possible future effort would be that the authoring tool connects to a webcam. The webcam could be used to record sign language videos while a scenario is being designed. A Deaf person or any person that can sign fluently could do this. The recorded sign language videos could also be edited within the authoring tool and then added to the scenario being created. Once the scenario is complete, it can be installed to a mobile phone, together with the recorded signed language videos, and be made available for use immediately.

Acknowledgment

We thank the Deaf Community of Cape Town for their involvement. Thanks also to George Ng'ethe, Marshalan Reddy, and Edwin Blake at UCT for their collaboration on this project. We also thank Telkom, Cisco, Aria Technologies and the THRIP (Technology and Human Resources for Industry Partnership) initiative of the South African Department of Trade and Industry for financial support via the Telkom Centre of Excellence (CoE) programme. THRIP funding (project TP13072623839) is managed by the National Research Foundation (NRF). Any opinion, findings and conclusions or recommendations expressed in this material are those of the authors and therefore the NRF/THRIP does not accept any liability in regard thereto.

References

[1] Blake, E., Tucker, W., & Glaser, M. (2014). Towards communication and information access for Deaf people. SACJ, (54), 10–19

[2] Motlhabi, M. B., Glaser, M., Parker, M., & Tucker, W. D. (2013). SignSupport: A Limited Communication Domain Mobile Aid for a Deaf patient at the Pharmacy. In R. Volkwyn (Ed.), Proc. *SATNAC* (pp. 173–178). Stellenbosch, South Africa; Pretoria: Telkom.

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

[4] Mutemwa, M., & Tucker, W. D. (2010). A mobile Deaf-to-hearing communication aid for medical diagnosis. In D. Browne (Ed.), Proc. *SATNAC* (pp. 379–384). Stellenbosch, South Africa; Pretoria: Telkom.

[5] Chininthorn, P., Glaser, M., Freudenthal, A., & Tucker, W. D. (2012). Mobile Communication Tools for a South African Deaf Patient in a Pharmacy Context. In P. Cunningham & M. Cunningham (Eds.), Proc. *ST-Africa*. Dar es Salaam, Tanzania; Dublin: IIMC International Information Management Corporation.

[6] Motlhabi, M. B., Glaser, M., Parker, M., & Tucker, W. D. (2013). SignSupport: A Limited Communication Domain Mobile Aid for a Deaf patient at the Pharmacy. In R. Volkwyn (Ed.), Proc. *SATNAC* (pp. 173–178). Stellenbosch, South Africa; Pretoria: Telkom .

[7] Gorman, W., Kniffin, B., & Oster, S. (2001) Web page authoring tool. *U.S. Patent Application 09/917,435*.

[8] Pressman, R. S. (2005). Software engineering: a practitioner's approach. McGraw Hill, 306-307.

[9] Mace, R. (1997). What is universal design. The Center for Universal Design *at North Carolina State University*. Retrieved Retrieved November, 19, 2004.

[10] Shneiderman, B. (2003) Designing the user interface. Pearson Education.

[11] Motlhabi, M. B., Tucker, W. D., Parker, M. B., & Glaser, M. (2013, December). Improving usability and correctness of a mobile tool to help a deaf person with pharmaceutical instruction. In Proc. DEV (Article 13). ACM.

[12] Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., & Yergeau, F. (1998). Extensible markup language (XML). *World Wide Web Consortium Recommendation REC-xml-19980210. http://www. w3. org/TR/1998/REC-xml-19980210*, 16.

[13] Hayes, G. R. (2012). Taking action in your research. *Interactions*, 19(4), 50–53.

[14] Jaspers, M. W. (2009). A comparison of usability methods for testing interactive health technologies: methodological aspects and empirical evidence. *International journal of medical informatics*, *78*(5), 340-353.

[15] Tongco, M. D. C. (2007). Purposive sampling as a tool for informant selection. *A Journal of Plants, People, and Applied Research, Vol. 5, pp. 147-158.*

Lindokuhle S. Duma received a BSc Honours in Computer Science in 2013 from the University of the Western Cape (UWC) and is presently studying towards an MSc with Bridging Applications and Network Group (BANG) at the same institution. His research interests include Deaf communication tools, authoring tools and cross-platform solutions. Duma is a Telkom bursar.

Prangnat Chininthorn received her Master's in 2011. She carries on her work to improve communication problems between Deaf and hearing people in health context through her PhD studies in the collaboration of Delft University of Technology and University of the Western Cape.

Meryl Glaser has an MSc in Human Communication specializing in Deaf People from the City University in London. Her research interests include Deaf education, communication, SASL and Deaf adult literacy, both text and ICT.

William D. Tucker is an Associate Prof. in Computer Science at UWC and leads the BANG research team there. His main research interests include community-based ICT4D projects and network technologies.