Abstractions for designing and evaluating communication bridges for people in developing regions

William D. Tucker
University of the Western Cape
Dept. of Computer Science
Bellville, South Africa
+27 21 959 2516
btucker@uwc.ac.za

Edwin H. Blake
University of Cape Town
Dept. of Computer Science
Cape Town, South Africa
+27 21 650 3661
edwin@cs.uct.ac.za

ABSTRACT

This paper describes two novel abstractions that help software engineers work in developing regions to align social and technical factors when building communication systems. The abstractions extend two concepts familiar to engineers of computer networks and applications: the Open Systems Interconnect stack for design, and Quality of Service for evaluation. The novel nature of the abstractions lies in how they help cultivate awareness of socio-cultural and technical issues when designing and evaluating communication bridges in the field. Advantages of the abstractions are that they can be understood easily by software engineers, they aid communication with beneficiaries, and can therefore facilitate collaboration. The paper makes an argument for these socially aware abstractions, describes the abstractions in detail, provides examples of how we used the new abstractions in the field and then gives practical guidelines for how to use them. The simple nature of the new abstractions can help software engineers and end-users to work together to produce useful information technology based communication systems for people in developing regions.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements /Specifications—*Elicitation methods*; H.5.2 [**HCI**]: User Interfaces—*User-centred design*

General Terms

Design, Measurement, Human Factors, Documentation

Keywords

Design and evaluation of applications, Participatory methods and user-centred design, User interfaces and accessibility for low-literacy populations, Information and communication technology for development.

ACM DEV'10, December 17–18, 2010, London, United Kingdom.
Copyright 2010 ACM 978-1-4503-0473-3-10/12 ...$10.00.

1. INTRODUCTION

When working in developing regions, network system and application software engineers have to consider many more issues than those captured by the common abstractions for guiding design and evaluation of networks and their applications, namely the Open Systems Interconnect (OSI) stack and Quality of Service (QoS), respectively. To help software engineers to make the necessary transition we introduce two novel abstractions, the Softbridge Stack and the notion of Quality of Communication, to complement OSI and QoS, and we show how they may be used. While these new abstractions are closely related to OSI and QoS, they have to sacrifice some of the features of the originals. For example, there is not necessarily a clear one-way dependence between adjacent layers in the Softbridge stack. Software engineers working in the emerging field of Information and Communications Technology for Development (ICT4D) have come to accept that rigid guidelines can no longer be given; just as researchers in the field have had to accept that qualitative accounts are frequently more appropriate and valuable than quantitative results.

The new abstractions described in this paper help software engineers to frame ICT4D work within a socio-cultural deployment context. The abstractions also help software engineers to talk to end-users to cultivate a mutual awareness of technological and socio-cultural factors. Software engineers and end-users can then collaboratively produce communication bridges for people in developing regions. We prefer the term *communication bridge* to communication system to emphasize the diversity of social and technical factors that have to be addressed and connected in developing regions. We therefore argue the need for socially aware abstractions in ICT4D. Our paper provides examples of using the new abstractions based on two case studies which are drawn from our ongoing projects over a period of seven years.

The ICT4D agenda has been embraced to broaden the scope of Computer Science with particular attention given to issues of power supply, networks, end-user devices and user interfaces [4]. Challenges within the ICT4D arena are simultaneously technical, environmental and cultural [5]. Technologists operating in this growing field are faced with difficulties on how to measure success and failure [14], how to choose appropriate technologies [1, 12] and design user interfaces [9]. A common thread is that the alignment of technical solutions with social underpinnings determine whether a given ICT4D solution is actually used or not. Thus any ICT4D effort must grapple with the interrelationships between social and technical issues. Most technologists are better equipped to deal with technical issues than social ones, and therefore many technological solutions end up in the field without being used due to social rather than technical reasons.

The social, or 'soft', issues within ICT4D projects can be examined with Monitoring and Evaluation (M&E) tools. We argue that such tools can also be used within a design context, while understanding that design is not the intended use of such instruments [26]. There are several comprehensive M&E tools available, e.g., the Universal Access Wheel [17] and Outcome Mapping [11]. We chose to concentrate on bridges.org's Real Access/Real Impact (RA/RI) criteria [6] for several reasons. Firstly, their criteria were compiled by examining a wide array of ICT4D projects across the globe. Secondly, this non-governmental organization (NGO) was based in Cape Town, South Africa, and we were therefore in a favourable location to interact directly with them, and were able to employ their consultation skills on our projects. Lastly, we found that the issues raised by similar M&E tools could generally be captured by the RA/RI criteria. Table 1 includes a brief overview of RA/RI. These criteria are meant to evaluate ICT4D projects after completion and not during development. We must be clear that we appropriated RA/RI for the purpose of design. The organization bridges.org disbanded several years ago and RA/RI is no longer being revised. At some point it may be advisable to adapt the criteria, but at present RA/RI remains a useful checklist for familiarizing the ICT4D software engineer with social issues.

As software engineers working in the ICT4D space, we are interested in communication networks and their applications. Students of networking are not given socially aware tools in formal academic environments [2]. These students face significant challenges in becoming what Heeks [14] calls a 'hybrid' — a person capable of aligning technical and social issues. Networking is taught with tacit assumptions concerning socio-cultural issues that emanate from the textbooks and their abstractions and tools. The abstractions taught for computer networking are not situated within the social or technological context of developing regions. Even in tertiary institutions in such regions, network and software engineers are trained with traditional design abstractions. This paper argues that traditional networking abstractions serve the ICT4D software engineer well for the design and evaluation of a very select range of *technical* issues. However, they fail to offer guidance on including *social* factors that are so important in the ICT4D arena; nor do they encompass a broad enough view of related technical and environmental issues (such as power provision). We address this deficiency by presenting novel abstractions that extend traditional notions of network design and evaluation with socio-cultural awareness.

The rest of the paper is organized as follows: Section 2 argues in more detail the limited applicability of traditional networking abstractions for ICT4D. Section 3 introduces and describes two novel socially aware abstractions to fill that gap between technical and social ICT4D issues. Section 4 provides examples from two case studies of how we used these new abstractions. Section 5 gives practical guidelines on how to use the novel abstractions in the field, and Section 6 offers concluding remarks.

2. COMMUNICATION ABSTRACTIONS

Abstractions help network and software engineers build and evaluate complex systems made of parts, by hiding complexity and details within those parts. Abstraction hides details from end-users with a user interface, and from other software engineers with Application Programming Interfaces (APIs). The OSI stack is typically taught in undergraduate networking courses. QoS is the standard method of evaluating networks and their applications in industry. This section critiques the OSI reference stack and QoS with respect to using them to design and evaluate communication bridges in developing regions.

Table 1: Real Access/Real Impact criteria [6]

Physical access Is the technology available and accessible to people and organizations?
Appropriateness Is the technology appropriate to the local needs and conditions of the community, and how people need and want to put technology to use?
Affordability Are the technology and services affordable for local people?
Human capacity and training Do people have the training and skills necessary to use the technology effectively? Do they understand how to use the technology and its potential uses?
Locally relevant content Are there locally relevant content and services available to people through the technology, especially in terms of language?
Integration to daily life Is technology use an additional burden to the lives of people, or is it integrated into their daily routines?
Socio-cultural factors Are people limited in their use of the technology because of their gender, race, age, or other socio-cultural factors?
Trust in technology Do people have confidence in technology use, and do they understand the implications of the technology they use? Do people understand issues like privacy and cyber crime?
Local economics Will the local economic environment sustain long-term technology use?
Macroeconomics Does the macro-economic environment in the country or region, such as issues like deregulation, investment, or labour affect technology use?
Legal and regulatory Do laws and regulations affect the use of technology?
Public support and political will Do people support the widespread use of technology in their communities? Do they understand the links between local technology use and the government's overall political strategies to promote ICT for development? Does the government have the political will to drive needed change?

2.1 Design Abstraction

The OSI stack is the most widely taught abstraction for organizing and hiding detail for computer networks and their applications. However, the OSI stack is an idealized reference model and the condensed four-layer TCP/IP stack is more often used in practice [7][22] (see Figure 1).

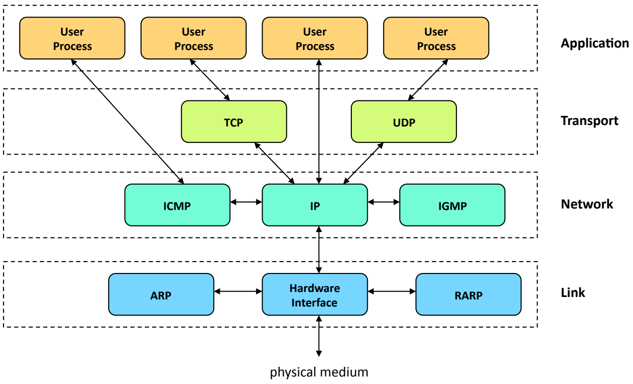

Figure 1: TCP/IP stack: This stack is a condensed version of the 7 layer OSI stack. Several common protocols are shown in the bottom three layers that parallel the OSI stack.

Each layer in a networking stack is responsible for different types of network functionality [22]. Only the bottom-most layer sends bits over a physical medium. Peers on either side of a network connection communicate at a given layer. Entities within a layer or in adjacent layers communicate with APIs, and entities within any given layer can be implemented differently at different hops in a network providing the APIs remain aligned. In both the OSI and TCP/IP stacks, the Application layer is not responsible for networking; lower layers handle all networking operations. The actual user application functionality, including its user interface, is undifferentiated within the topmost layer. The API into the transport layer is where the application calls upon network functionality. For the network software engineer, that can be as far as the abstraction goes since application functionality is someone else's responsibility. In practice, this effectively means that the networking stack extends up to, but does not really include, the application that an end-user interacts with. This point is important because people use communication systems, and ultimately these end-users decide whether the systems are acceptable or not.

OSI and TCP/IP design stacks are inadequate for ICT4D, firstly, because they exclude the user space. The stacks, and consequently their users, make implicit and tacit assumptions about users and their environment. Heeks [14] argued that the communication network and/or application designer tacitly assumes that the technical needs and social environment of the end-user are the same as the designer's, almost certainly entailing a misalignment between design considerations and ICT4D scenario realities. Heeks [14] called this the 'design-reality gap'.

Brewer et al. [4] argued that technologists working in an ICT4D landscape must also consider technical, environmental and cultural factors that are often foreign to a traditional or corporate technology mindset. Thus within ICT4D, the network designer has to address issues like unreliable power and intermittent connectivity, and the application designer must accommodate social issues like the illiteracy of users and the sharing of handsets, with special attention devoted to devices and the user interface. Technological considerations must therefore be contextualized within the socio-cultural realities of ICT4D. There are many notable efforts within the computer science literature to address these issues. ICT4D-specific design considerations are typically portrayed in the literature as lists of lessons learnt and recommendations [4, 5, 8]. However, these lessons have thus far not been formally factored into a design abstraction for computer scientists working in the ICT4D domain.

2.2 Evaluation Abstraction

Evaluation abstractions are implicitly tied to design abstractions. Thus a similar situation holds for ICT4D evaluation. Communication networks and their applications designed with the OSI and TCP/IP stacks are commonly characterized with an abstraction called QoS that measures technical performance at various levels in the OSI stack, usually Layers 2-4. Network evaluators conduct carefully controlled tests to determine an end-user's perception of network and application performance. The ITU provides the dominant QoS definitions for real-time voice and video communication networks and their applications [15].

If the design of networks and their applications must consider socio-cultural issues, then so must evaluation. Thus even though QoS explicitly targets an end-user's perception of a system, that perception is framed within the evaluators' assumptions. Whereas traditional design abstractions suffer from Heeks' [14] 'design-reality' gaps in an ICT4D context, QoS suffers from corresponding 'evaluation-reality' gaps. Traditionally, QoS is used to evaluate user perception of technical network performance objectively, so that communications can be costed accordingly. Evaluators in a traditional networking or telco context, just like designers, tacitly assume performance expectations and social factors. One consequence is that QoS is restricted to evaluation of real-time voice and video.

ICT4D considerations expressly include non-real-time communication networks and applications in all forms of media [4, 18]. Note that QoS does not address text-based communication at all, or the use of images rather than video. ICT4D applications may also require voice and video, yet both are much more demanding of bandwidth than text and asynchronous communication. The validity of the traditional QoS mindset is called into question in both ICT4D settings [19, 20] and non-ICT4D settings [3] because allocating more bandwidth is avoiding the problem rather than solving it. In ICT4D settings particularly, bandwidth is a scarce commodity and even less of a solution. Therefore, objective evaluation of ICT4D networks and their applications is divided between traditional QoS evaluation where appropriate, like measuring packet loss [21], and measuring what users actually expect from a given system, e.g., how they use the Internet [10]. Thus, where QoS is fundamentally technical and/or centred on a particular kind of user, the ICT4D context requires a more socially situated evaluation.

The limitations identified above lead us to the conclusion that there is more to the design and evaluation of ICT4D than provided by the OSI and TCP/IP stacks, and QoS. Traditional design stacks address many technical issues, however, socio-cultural, human computer interface (HCI) and application-oriented factors of ICT4D are outside the purview of those abstractions except as implicit assumptions. OSI and TCP/IP stacks are only applicable to the technical network considerations. The next section describes a socially aware design stack that includes technical, socio-cultural and environmental factors.

In a similar fashion, traditional QoS evaluation of networks and their applications are tightly coupled to a particular kind of user's expectation of technical network performance. However, as for design, there is more to ICT4D communication system evaluation than a user's perception of latency, jitter and packet loss. Social factors may be much more important than an individual user's perception and/or technical issues for take-up of ICT4D solutions in the field. In other words, networks and their applications may work perfectly well in the laboratory, and even in the field, yet socio-cultural factors will play a significant role in whether they are used or not. Furthermore, real-time communication associated with QoS is neither always needed nor feasible.

3. SOCIALLY AWARE ABSTRACTIONS

This section presents two novel abstractions that are intended to supplement OSI and QoS abstractions by providing mechanisms to cultivate an alignment of technical and socio-cultural issues. The abstractions are primarily meant for software engineers, however it is intended that the constructs can also be discussed with end-users. We argue that software engineers can become aware of socio-cultural issues that affect various parts, or layers, of a technological solution by using a design stack that includes social as well as technical factors. It follows that end-users must also become aware of technical issues, albeit at different levels of abstraction. If end-users are to take part as co-designers, they must also be able to understand a high-level approach to technology design that includes their socio-cultural domain, a domain in which they are the experts.

A similar two-way approach can be applied to evaluation. Evaluators measure objective performance and user perception of that performance. However, the analysis of data needs to situate and contextualize the socio-cultural milieu where the ICT4D solution is to be used. Furthermore, evaluation needs to be performed by both technologists and end-users in order to feed back into the design process to produce solutions in an iterative fashion. Therefore there is one new abstraction each for design and evaluation. Section 3.1 describes the Softbridge stack, a socially aware companion to the OSI stack and Section 3.2 describes Quality of Communication (QoC), a socially aware companion to QoS.

3.1 Softbridge Stack

We take the view that communication bridges are *soft* from both social and technical perspectives. From a social perspective, soft issues are people issues; for example, a person's reluctance to change and use a new or unfamiliar technology to establish a connection to someone else, or a government's constraint on the use of a technology like Voice over Internet Protocol (VoIP) that makes the use of a VoIP bridge illegal. From a technical perspective, communication bridges are soft because they are mostly constructed with software. Software is inherently malleable and changeable. VoIP also offers a good example of malleable software that can easily be customized and adapted for different uses, e.g., real-time communication, instant voice messaging and push-to-talk. Because both people and technical issues are soft, we call our novel design abstraction the Softbridge stack, in that it encompasses the soft technical and social aspects involved in providing soft ICT4D bridges between people. The Softbridge stack consists of seven layers: power, network, device, media, temporality, user interface and people (see Figure 2). On the surface, the Softbridge stack looks like a generalized technology stack connecting people at the top. However, each layer involves both technical and social issues. Note that all of the examples used in what follows to describe social issues of a particular technical Softbridge layer can be connected to a particular RA/RI criterion (see Table 1). Thus, M&E tools (RA/RI in this case) can be applied in order to help understand design issues at all layers in the Softbridge stack.

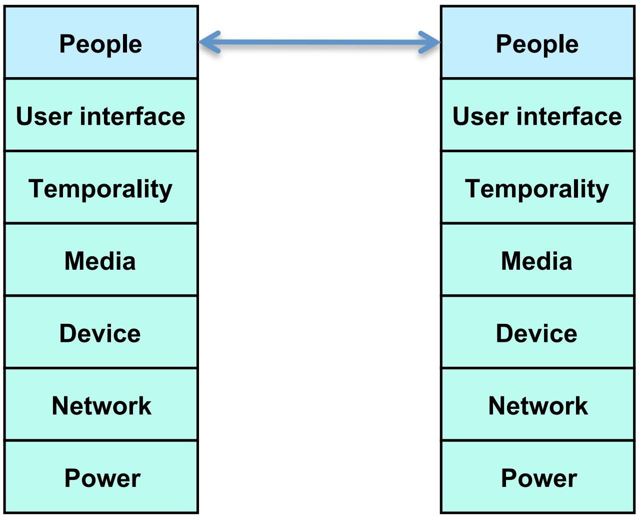

Figure 2: Softbridge stack: This stack is meant to compliment the OSI and TCP/IP stacks (Figure 1). The relationship is further illustrated by Figure 3. The Softbridge stack includes a wider array of factors found in ICT4D scenarios, especially the people layer concerning soft issues.

The Power layer deals with issues like poor power provision, a common environmental factor in developing regions where power is frequently disrupted, often for long periods of time. No technology can operate without some form of power. Power layer considerations include battery size/duration, power management techniques, and can include design decisions that bridge between different forms of power sources on either end of a connection. There are also social aspects. For example, in rural areas many people with mobile phones charge the device at a nearby shop because they do not have power, mains or solar, at their home. Theft of power equipment, and of power itself, are also common occurrences in developing regions. The goal of this layer, then, is to ensure that both ends have some form of power to enable communication, although as will be discussed below, not necessarily at the same time.

The Network layer involves networking equipment and protocols such as those most commonly associated with the OSI and TCP/IP stacks. ICT4D networks are very often wireless, and network bridges are easily dealt with by software engineers. However, a typical end-user's understanding of networks often stops at the network's name, e.g., WiFi, 3G or Bluetooth. An end-user may not need to know how a particular network works, but should understand how and why to use various types of networks in different situations. For example, to save money and obtain decent QoS and/or QoC, an end-user should text over a GPRS connection and rather use a WiFi network for VoIP. The Network layer provides bridges between various types of networks to enable this to happen, and the software engineer can still use OSI and TCP/IP stacks, and QoS, to build and evaluate these bridges, respectively. The relationship between the Softbridge and OSI stacks is visualized in Figure 3.

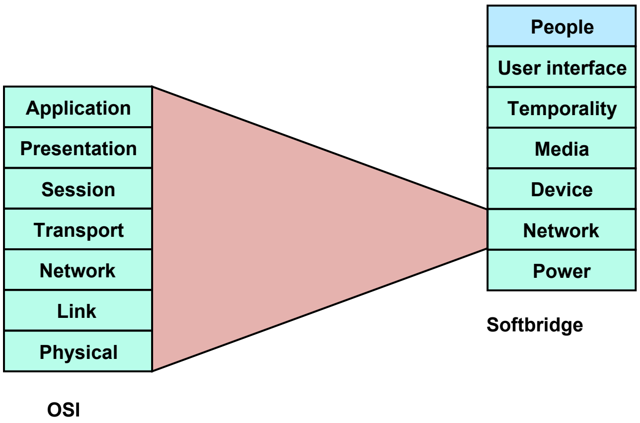

Figure 3: Softbridge's relationship to OSI: The familiar OSI stack layers are concentrated into the network layer of the Softbridge stack. This is low-level network functionality called upon by an application running on a given device.

The Device layer deals with bridges between the wide varieties of end-user devices that can be deployed. Mobile devices are particularly useful in the ICT4D context. With reference to the aforementioned layers, mobile phones have long-life batteries and can often connect to different networks. Design considerations at the Device layer include choosing an appropriate device for an end-user. The considerations are not always technical. As examples, a smart phone may be an ideal device for a given ICT4D solution, yet be too expensive to disperse to a large underemployed population, handsets can be shared by multiple users (and so should allow privacy settings), users might want to use multiple SIM cards, and devices like netbooks may be ideal for technologists, yet too complicated for computer-illiterate users. The communication applications and the networks they use for transport therefore need to work across a wide variety of end-user devices.

The Media layer is where the application space begins, and of course, must use APIs available on a given device. This layer includes support for multiple media modalities such as text, voice, images and video. Different forms of information and communication can make use of all types of media. Hence, the software engineer makes design decisions concerning protocols, hardware support and APIs at this layer. Media bridges may be homogeneous, e.g., VoIP or heterogeneous, e.g., multiple forms of media in a mash-up like Facebook, and can also be adapted, e.g., text to speech for illiterate users. End-user concerns feature prominently here because their needs determine the types of media required especially regarding content in a local language.

The Temporality layer[1] entails considerations of providing any type of media across a continuum of synchronous, asynchronous and semi-synchronous transfer. These temporal modalities can also be adapted, e.g., a real-time VoIP call can be transformed into asynchronous voice mail when a called party is not available. The temporal nature of networks and their applications must accommodate the temporal rhythms already in place without ICT4D, for example, the slow paced nature of 'Africa time' that places a premium on personal interaction. In addition, environmental factors such as poor power and remoteness dictate best-effort in the extreme, with semi-synchronous and asynchronous communication applications, e.g., Instant Messaging, web browsing and email that work equally well over real-time and delay-tolerant networks, no matter the size of the bandwidth pipe.

The User Interface layer puts a wrapper on top of lower levels in the Softbridge stack in order to present networks and their applications to end-users. For applications, at least, the user interface is indeed a component of the application. The user interface to a network can also work its way through an application, e.g., a dialog to use a packet data connection on a mobile phone. Since the user interface is what the end-user sees, as it hides underlying complexity, standard HCI techniques can be employed to work at this level. However, we argue that even these techniques need to be cultivated and adapted with socio-cultural awareness in order to be effective [2].

The People layer represents communication bridges between people. Unlike in the OSI and TCP/IP stacks, the ultimate communicating entities in the Softbridge stack are people, not applications. Thus the topmost layer in the Softbridge stack is a People layer. This is perhaps the most important and most difficult layer to design for, since in many ways communication between people manifests a socio-cultural milieu that needs to be understood by software engineers in order to design networks and applications for it. However, the People layer is what end-users are experts of, whether they know it or not. So, the end-users are the primary vehicles of generating user requirements for ICT4D solutions, as they rightly should be.

3.2 Quality of Communication

We devised another novel abstraction called Quality of Communication, or QoC, to evaluate ICT4D communication bridges. Whereas QoS measures user perception of performance metrics at layers in the OSI stack, QoC evaluates layers within the Softbridge stack with ICT4D M&E tools (e.g. RA/RI, see Table 1) to factor in socio-cultural factors to help evaluate each Softbridge layer. This use of RA/RI for evaluation is more in line with the original intent of M&E, and is primarily qualitative. Just as QoS analysis results in tweaking design of OSI layers, QoC analysis results in tweaking design requirements and solutions at Softbridge layers.

QoC is based on the Softbridge stack, therefore technologists and end-users familiar with Softbridge concepts are also able to participate in the QoC evaluation process. QoC reaches beyond the end-user to include the user's social environment in the appraisal of communication quality rather than only the technical characteristics and user-centric fac-

[1] The term "Temporality" could also be called "Time" or "Synchrony".

tors that QoS addresses. QoC is more concerned with an assessment of how ICT enables mutual "intelligibility" [23] between communicants within a social context, whereas QoS is more concerned with system performance [3]. Thus, QoC takes the view that while QoS is a 'nice to have', it is often not necessary. QoC is rather more qualitative, using M&E criteria to examine how computer mediated communication at each Softbridge layer functions with respect to social mechanisms and challenges. Common methods include semi-structured interviews, focus groups and participant observation.

From a quantitative perspective, QoC is therefore more challenging to measure than QoS because as with Softbridge, QoC is based on awareness of socio-cultural factors. Mechanically, however, some aspects of QoC can be measured in a similar fashion to QoS. There are some significant differences, however. For example, instead of measuring latency, jitter and packet loss at the packet level in a particular OSI layer, QoC could measure similar characteristics in terms of complete messages. QoC measurement could count the number of messages sent and received, lost, how long they are, and the latencies between messages sent and received. The temporal aspect is a major difference between QoC and QoS because QoC measurements are at a much larger scale, e.g., in the order of days instead of milliseconds. These quantitative measurements can then be triangulated with qualitative data collected and analyzed together with end-users. The social and unconventional temporal aspects of QoC encourage out-of-the-box thinking to understand why ICT4D solutions get used or not regardless of how well or how poorly the actual applications perform from a QoS perspective.

For example, a real-time VoIP application might provide acceptable QoS in the laboratory and still not get used in the field where intermittent and/or low bandwidth cannot support that application. Then, to take this example farther into the social space, if communicants cannot find the time to establish real-time communication because of various reasons, then asynchronous communication with huge delays becomes the only viable option. Asynchronous communication might offer very poor QoS but result in good QoC!

Thus, QoC offers a way to understand, and hopefully avoid, 'evaluation-reality gaps' just as using the Softbridge stack offers a way to help avoid Heeks' [2] 'design-reality gaps'. The design and evaluation processes are often complementary and overlapping. Together, the Softbridge stack and QoC can help software engineers work with end-users to cultivate mutual awareness to achieve an alignment of technical and social factors for an ICT4D solution. The following section provides some specific examples from two case studies that we conducted in the field.

4. EXAMPLES FROM CASE STUDIES

We evolved and employed the Softbridge stack and QoC abstractions by working with two ICT4D cases in South Africa: one on Deaf telephony and another on rural telehealth. This section briefly introduces each case study and then walks up the Softbridge stack from the lowest layer to the highest layer to provide examples of socially aware design and evaluation made at each layer. Complete details on both case studies can be found in [25].

4.1 Examples from Deaf Telephony

The Softbridge concept originated from a Deaf telephony case study in which we wanted to build semi-automated bridges between Deaf and hearing people with an NGO called DCCT, the Deaf Community of Cape Town. The capital 'D' in Deaf denotes the use of sign language to identify a sign language speaker with a particular socio-cultural group. Our Deaf participants use South African Sign Language (SASL) and had some limited experience with a locally designed text-based telephone called a Teldem. Our initial efforts attempted to convert text-to-speech (TTS) from that device, relay the speech to a hearing person with a normal or mobile handset using VoIP protocols like H.323 and Session Initiation Protocol (SIP), and then perform the reverse direction with Automatic Speech Recognition (ASR). We went on to explore many other kinds of communication applications for Deaf users, mostly concerning video for sign language support on computers and mobile phones. Some brief examples below illustrate the use of Softbridge and QoC.

Power: There were no explicit design options made for power. The power availability in Cape Town is comparable with most developed regions in the world. We did, however, observe that WiFi appeared to consume less battery than GPRS on mobile handsets.

Network: We devised various bridges between PSTN, GSM and Internet Protocol networks using H.323 and SIP gateways and clients. End-users were not aware of any of these protocols at first, yet through participation in the project became aware of different types of networks. We provided broadband Internet at the Deaf community centre and monitored its usage. After several years, Internet usage steadily increased and the NGO assumed payment for it. This told us that the Deaf people found value in Internet access. Soon a small group of Deaf users began using data with both 3G and WiFi on their mobile phones. None of them had done that prior to our project. Note that South Africa has some of the most expensive Internet in the world, especially over cellular networks, e.g., R 2 (US 25 cents) per megabyte.

Device: The Deaf users did not like using the Teldem for many reasons both technical and social [13]. We therefore developed subsequent prototypes of voice relay with instant messaging APIs on computers, including Jabber and SIMPLE. We also tried out an SMS, or texting, interface on a mobile phone for the Deaf user to send and receive text to the relay system because many Deaf people at the time were familiar with SMS. In order to get Deaf people to use the application on a computer, we first had to train them with basic computer literacy. Limited computer usage was further compounded by the fact that none of the Deaf users owned computers at home, or even used them at work. We provided six computers at the Deaf community centre. However, transport to the centre even to use the computers was an issue due to the expense, limited hours and difficulty of using public transport in Cape Town. We came to realize more and more that mobile phones offered ways to get around these social factors linked to computer access and literacy. However, mobile phones with advanced capabilities are beyond the financial reach, and literacy, of most people associated with DCCT.

Media: The relay prototypes were concerned primarily with text and voice. Text became a serious concern because Deaf users are not necessarily literate in any written languages which are not their first language (that is, SASL).

While they frequently SMS'd each other, they were painfully aware of their poor English spelling and grammar that the hearing users would have translated into speech. Thus, they would use poor English with each other, not necessarily the accepted SMS shorthand, yet not want to use a written language with hearing people. Through this, we came to understand that Deaf users wanted to communicate in sign language and would therefore require very high quality video. One significant problem with video for Deaf people is the cost, particularly on mobile networks. In addition our Deaf users frequently do not have phones that support high quality of video capture or playback unless we provide such phones.

Temporality: The use of TTS and ASR engines (or human relay operators when we found that ASR did not perform well with South African English accents) incurred very high latency. Thus what would have otherwise been a real-time relay became semi-synchronous at best. At the time we started considering video, even Internet-based video did not provide enough video quality for sign language comprehension. The situation was, and remains, worse for mobile phones. Therefore we began investigating the use of asynchronous video, as opposed to video streaming, that offers better sign language comprehension at the expense of higher latency.

User interface: To deal with the two-way conversion latencies, we used 'isTyping' interface techniques to let each type of user know when the other user was talking or typing. For example, while a Deaf user was keying in a message on a computer keyboard, we would play music to the hearing user. The computer user interface in general was problematic to Deaf users simply due to lack of familiarity. When we started considering applications on mobile phones, we also found that advanced mobile literacy, such as web browsing and instant messaging, was also lacking.

People: When it came to linking up people, we had to adjust the aims of the project by considering that Deaf people needed to communicate with other Deaf people as much if not more than with hearing people. We also found that the Deaf community shunned the use of popular communication technologies like Facebook and MXit[2] because of negative tabloid coverage. We believe improved online privacy skills could help overcome this aversion to trying out new and exciting communication technologies.

4.2 Examples from Rural Telehealth

During the same time period, we also conducted a rural telehealth case study in the remote Eastern Cape of South Africa, mostly with the help of an NGO called Transcape. The rural WiFi network was, and remains, successful yet the telehealth aspects were not. We attempted to provide communication bridges between nurses at a remote clinic and doctors at a rural hospital. We provided wireless networks and a custom application to do this, and they were not used primarily due to socio- cultural factors. In two different districts, we found most of the doctors to be foreigners working on contracts of several years, and also that none of the nurses were from the villages they served. The clinic and hospital staff also reported to different management structures. Most significantly, the nurses and doctors were overwhelmed with day-to-day duties. ICT was a burden for them, even though it offered opportunities to improve service.

Power: We dealt with unreliable rural power sources by running the entire network off of 12v deep cycle batteries charged by either mains or solar. End-user devices could also be charged from those batteries as much as possible, e.g., mobile phones with car chargers. We deliberately chose the laptops with the largest batteries possible, and also provided backup batteries (that were, incidentally, not used). Solar panels and batteries were sometimes stolen so we positioned equipment in secured structures or at the homesteads of village headmen.

Network: The network was the most successful part of the project because we based the architecture and firmware on the work of collaborators (www.rurallink.co.nz), using a combination of 2.4 and 5GHz links of up to 15km. We measure these networks with standard QoS metrics and found acceptable performance, e.g., ping times, roundtrip times and acceptable latencies. At one point, the network was rebooted only twice in four years. We continue to devote significant time educating end-users on the differences between WiFi and cellular networks, e.g., the range of the each network and why the WiFi link disappeared when a user left the village.

Device: We chose laptops and mobile phones over computers in order to achieve longer operation times due to the power issues. We taught users how to manage power on the laptops, e.g., adjusting sleep modes and screen brightness. We came to focus more on mobile phones because the laptop peripherals were cumbersome to end-users, e.g., microphone headsets caused the users grief like not being able to hear a call when in another room, and the transfer of images from digital cameras involved too many steps. Mobile phones unified many peripheral features in one device. We also dedicated trainers to help nurses and doctors with computer literacy. The end-users were much more comfortable with mobile handsets and we chose smart phones that resembled more normal handsets.

Media: To help nurses and doctors communicate about patients, we provided text, images, voice and video. We devised a fully open source multi-modal instant messaging application based on patient cases. A smart phone enabled us to combined capture and playback of all modalities on a single device. We also verified acceptable QoS on the real-time voice and video components over the wireless networks.

Temporality: Two factors in particular caused us to adapt a mixed synchronous and asynchronous approach to all forms of communication media: poor power for laptops and the hectic schedule of the participants. Thus, we emphasized the asynchronous aspects with end-users because we hoped that it addressed those two situations simultaneously. In actuality, end-users rarely used either temporal mode despite telling us how helpful the application would be.

User Interface: We based the user interface on the idea of patient 'cases' that actually led to a rather complicated combination of instant messaging, email and real-time media interfaces. A mobile handset enabled the simplification of the interface by having it resemble SMS, something end-users were already familiar with. We did not address the fact that nurses and doctors had different home languages, as they communicated with each other in English, a second or third language for all involved.

[2] MXit is an extremely popular locally developed South African instant messaging system that runs on almost any mobile phone that supports data. For more information, see www.mxit.co.za.

People: Participants told us they liked having these communication options available yet actually rarely used the telehealth application to discuss patients. Thus, triangulating actual application usage statistics with user feedback revealed a mismatch that required factoring in the social context. For example, the nurses would not have used the system no matter how well it worked because as one of them explained, the system could be used to 'spy' on them to see if they were indeed in the clinic or not.

5. PRACTICAL GUIDELINES

Having presented two socially aware abstractions for design and evaluation, and some examples of using them in two case studies, this section suggests practical guidelines on how to use the Softbridge stack and QoC in practice. First, we make some comments on how to cultivate awareness of socio-cultural issues in ICT4D. Then, we conclude by giving suggestions for the design and evaluation processes with the new abstractions, respectively.

5.1 How to Cultivate Awareness

The best way for ICT4D software engineers and end-users to cultivate mutual awareness of each other's domains is to talk. Communication must be both within and between groups. To get software engineers talking about social issues, start ICT4D education early in tertiary education, especially at institutions in developing regions. We typically wait until the final year to give a course on ICT4D, if at all. A practical way to get people thinking about these issues is to workshop the tools. We were fortunate to be able to conduct regular evaluation workshops on student projects with bridges.org personnel in the early phases of this project. We continue to conduct Softbridge and QoC workshops without them. We also treat all interaction between/with stakeholders as opportunities for training, workshops, and networking. We emphasize both formal and informal interactions, and are not afraid to fraternize with participants. A particularly helpful way is to schedule informal meals with participants after a formal session because an informal setting combined with time to think about a workshop can lubricate open opinion sharing (especially when drinks are provided!). Finally, we try to use communication prototypes intended for end-users *with* end-users and each other as much as possible. That helps us debug systems and also determine what our end-users actually want to use the systems for.

5.2 How to Design ICT4D with Softbridge

Our approach to designing ICT4D with the Softbridge stack was to fuse networking and HCI software engineering techniques in an iterative process based on iterative cycles. There is a strong correlation between standard software development life cycle (SDLC) activities and action research (see Figure 4). The iterative approach to software design is widely accepted. We embraced action research due to its dual imperative of addressing both academic and community goals [16] that we found harmonious with ICT4D goals. The merits of action research are numerous and compelling [24].

The way to start using the Softbridge stack is to consider each layer in turn from the bottom up. Design and debug with network stacks always starts at the bottom and works its way up one layer at a time. For each Softbridge layer, appropriate M&E criteria, e.g. RA/RI, as a checklist to consider social issues at a specific layer. However, what one finds, as can be seen by the examples in the previous section, is that the layers frequently overlap. Relationships emerge between Softbridge layers, and with so many social issues to consider, one way to organize it all is to view a particular problem or set of problems as a matrix with the Softbridge stack on one axis and the M&E criteria on another. In practice, such as with twelve RA/RI criteria, handling textual information in such a large table becomes problematic. Also, not every M&E criterion is applicable at each Softbridge layer. It is better to keep to the Softbridge stack structure and identify appropriate social factors at each layer and then relate them to issues in other layers in iterative cycles. This approach is useful for both design and evaluation.

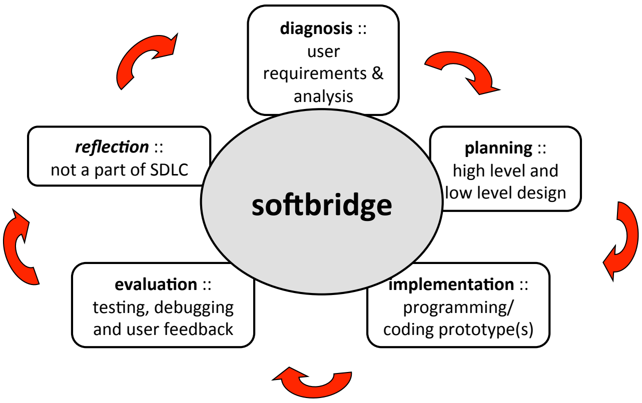
diagnosis :: user requirements & analysis
planning :: high level and low level design
implementation :: programming/ coding prototype(s)
evaluation :: testing, debugging and user feedback
reflection :: not a part of SDLC
softbridge

Figure 4: Softbridge in action: Action research stages appear in boldface corresponding to familiar stages of the software development life cycle. For example, high level and low level design in the SDLC can be mapped to the planning stage of action research. Reflection is explicit to action research and can be considered implicit to the SDLC.

Another way to use the Softbridge stack is to consider the layers as petals in a flower of overlapping interrelationships with people issues, identified via an appropriate M&E lens, at the centre (see Figure 5). This view also serves to link aspects at different Softbridge layers together. In this view, the flower's interdependencies are also applicable for both design and evaluation.

5.3 How to Evaluate ICT4D with QoC

Collect quantitative data, e.g., performance and usage statistics, by instrumenting the prototypes to do so. Collect qualitative data from interviews and focus groups with end-users, during the aforementioned workshops and informal gatherings and also by watching end-users in action. For effective QoC analysis, triangulate quantitative and qualitative data iteratively with deep reflection on M&E issues to cultivate awareness of socio-cultural and technical factors. Again, we chose to use bridges.org's RA/RI criteria and organized workshops, discussions and reflection around those criteria. That reflection can make use of either flower or matrix approaches to the Softbridge stack. Some more practical ways to conduct this analysis are similar to what was suggested in the previous section.

Conduct workshops with external reviewers (also known as "critical friends") where students present an analysis of

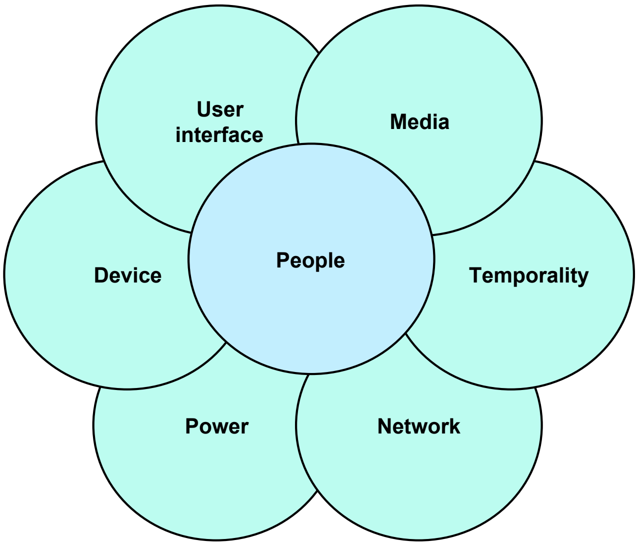

Figure 5: Softbridge stack as a multi-petaled 'flower': The overlaps are not meant to be restrictive, e.g., the user interface can also have dealings with the network layer as mentioned in the user interface portion of §3.1.

their work with the Softbridge stack and QoC. These workshops can also be conducted with end-users. Integrate feedback from both participants and external reviewers into reflection for the next cycle.

When in the field for long periods of time, conduct end-of-day focus groups, or if close to home, have a group session when done. Always ask developers and participants to write summaries after notable events/visits/trials to get ideas for design and evaluation in writing. The continual communication between developers and participants necessitates the use of plain *language* and terms. We advise our students to write their documentation 'for Mom' so that their mothers could understand what they are doing. We also stress the use of metaphors to explain concepts to other software engineers, intermediaries and to end-users.

Note that it is perfectly acceptable to apply OSI and QoS where appropriate for low-level technical design and evaluation only, e.g., designing or tweaking a protocol. Use the Softbridge stack and QoC to factor in social issues, to explain how and why technologies get used in ways not intended or when solutions do not get used even though they do actually function properly.

6. CONCLUDING REMARKS

The Softbridge stack and the notion of QoC are meant to help the ICT4D software engineer in two respects: firstly to unpack and consequently cater for both socio-cultural and technological factors, and secondly to serve as guidelines for talking with actual end-users with a view toward co-design and co-evaluation. Over the course of two long-term case studies, we have found that basic network infrastructure at the lowest layers in the Softbridge stack is the easiest part of ICT4D for the technologist to provide. We have also found that end-users in developing regions are willing to pay for and use this infrastructure. Low level infrastructure is also consequently the easiest to evaluate because the traditional abstractions studied in the classroom, like OSI and QoS, are indeed most applicable at the network layer. QoS metrics are also particularly easy to quantify. On the other hand, the qualitative social-cultural factors are much more challenging to understand. We used bridges.org's RA/RI criteria to help analyze such factors, but one could substitute any ICTD-oriented M&E tool. Regardless of the tool and its checklist, social 'soft' issues are the most difficult to identify and cater for. This is why we expressly worked with an NGO in each case study. Techno-savvy NGO members can be, and become, ideal hybrids of technical and social resources. We have found that long term iterative cycles are essential so that software engineers, NGOs and end-users can come together to address tough ICT4D problems.

The Softbridge stack is a purposefully leaky abstraction, leaving room for software engineers to weave together interrelated technical and social factors to build sustainable communication bridges. The QoC concept is consequently broad due to that leakiness. Because we have found that social factors predominate in the take up of a given bridge, we recognize the need for dedicated techniques to measure various aspects of QoC — ways that characterize or even quantify how a given social variable affects a technological bridge. The challenge remains that the wide array of socio-cultural factors interplay with each other, and consequently interact with technological choices and implementations. We hope that that the Softbridge and QoC abstractions represent steps toward characterizing those interrelationships in a format familiar to software engineers. Thus, Softbridge and QoC can help software engineers become aware of socio-cultural issues, and enable them to communicate with end-users to collaboratively address ICT4D challenges.

7. ACKNOWLEDGMENTS

The authors thank the staff and members of Deaf Community of Cape Town (DCCT) and Transcape.org for their participation in the case studies, as well as the Federation of Rural Coastal Communities (FRCC). Postgraduate students involved in both projects built the prototypes and conducted a great deal of the fieldwork: Jason Penton, Xolisa Vuza, Marshini Chetty, John Lewis, Tao Sun, Elroy Julius, Andrew Maunder, Wilson Wu, Gary Ma and Koos Looijestejn. The following colleagues were very helpful: Teresa Peters, Philipp Schmidt, Jennifer Huesler, Murray Pearson, Rudi Westerveld, Adinda Freudenthal, Nicola Bidwell, Gary Marsden and especially Meryl Glaser. The International Development Research Centre (IDRC) provided some financial assistance for the rural telephony project, and the South Africa Netherlands research Programme on Alternatives in Development (SANPAD) provided some financial assistance for both projects. We also thank Telkom, Cisco, Nokia-Siemens and the THRIP (Technology and Human Resources for Industry Partnership) initiative of the South African Department of Trade and Industry for financial support of both projects via the Telkom Centre of Excellence (CoE) programmes at the authors' respective universities. THRIP funding 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 does not accept any liability in regard thereto.

8. REFERENCES

[1] M. L. Best. The wireless revolution and universal access. In ITU, editor, *Trends in Telecommunications Reform*. International Telecommunication Union, Geneva, 2003.

[2] E. Blake and W. Tucker. Socially aware software engineering for the developing world. In P. Cunningham and M. Cunningham, editors, *IST-Africa 2006 Conference Proceedings*. IIMC International Information Management Corporation, 2006. Available from: `http://www.ist-africa.org/conference2006/default.asp?page=paper-repository`.

[3] A. Bouch, A. Kuchinsky, and N. Bhatti. Quality is in the eye of the beholder: meeting users' requirements for internet quality of service. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 297–304, New York, NY, USA, 2000. ACM. `doi:10.1145/332040.332447`.

[4] E. Brewer, M. Demmer, B. Du, M. Ho, M. Kam, S. Nedevschi, J. Pal, R. Patra, S. Surana, and K. Fall. The case for technology in developing regions. *Computer*, 38(6):25 – 38, May 2005. `doi:10.1109/MC.2005.204`.

[5] E. Brewer, M. Demmer, M. Ho, R. Honicky, J. Pal, M. Plauche, and S. Surana. The challenges of technology research for developing regions. *Pervasive Computing, IEEE*, 5(2):15 – 23, apr. 2006. `doi:10.1109/MPRV.2006.40`.

[6] bridges.org. Real access / real impact criteria. www.bridges.org, 2004.

[7] D. E. Comer. *Internetworking with TCP/IP: Principles, protocols, and architecture*, volume 1. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 4th edition, 2000.

[8] J. Donner, R. Gandhi, P. Javid, I. Medhi, A. Ratan, K. Toyama, and R. Veeraraghavan. Stages of design in technology for global development. *Computer*, 41(6):34 –41, jun. 2008. `doi:10.1109/MC.2008.203`.

[9] S. M. Dray, D. A. Siegel, and P. Kotzé. Indra's net: Hci in the developing world. *interactions*, 10(2):28–37, 2003. `doi:10.1145/637848.637860`.

[10] B. Du, M. Demmer, and E. Brewer. Analysis of www traffic in cambodia and ghana. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 771–780, New York, NY, USA, 2006. ACM. `doi:10.1145/1135777.1135894`.

[11] S. Earl, F. Carden, and T. Smutylo. *Outcome Mapping: Building Learning and Reflection into Development Programs*. International Development Research Centre, Ottawa, Canada, 2001.

[12] H. Galperin. Wireless networks and rural development: Opportunities for latin america. *Inf. Technol. Int. Dev.*, 2(3):47–56, 2005. `doi:10.1162/1544752054782420`.

[13] M. Glaser. A field trial and evaluation of Telkom's teldem terminal in a Deaf community in the Western Cape. In *Proceedings of Southern African Telecommunication Networks and Applications Conference (SATNAC)*, Somerset West, 2000. Telkom. (CD ROM publication).

[14] R. Heeks. Information systems and developing countries: Failure, success, and local improvisations. *The Information Society: An International Journal*, 18(2):101–112, 2002. `doi:10.1080/01972240290075039`.

[15] ITU-T study group 12. *Communications Quality of Service: A framework and definitions*. International Telecommunication Union, Nov. 2001. Recommendation G.1000 (11/01). Available from: `http://www.itu.int/rec/T-REC-G.1000-200111-I`.

[16] J. McKay and P. Marshall. The dual imperatives of action research. *Information Technology & People*, 14(1):46–59, 2001. `doi:10.1108/09593840110384771`.

[17] T. Oyedemi. Universal access wheel: Towards achieving universal access to ict in africa. *The South African Journal of Information and Communication*, (5):90–107, 2004.

[18] A. Pentland, R. Fletcher, and A. Hasson. Daknet: rethinking connectivity in developing nations. *Computer*, 37(1):78 – 83, jan. 2004. `doi:10.1109/MC.2004.1260729`.

[19] O. W. Purbo. Alternate strategy to deploy fixed wireless broadband services in developing countries. http://sandbox.bellanet.org/ onno/ict4d-roadmap/alternate-strategy-to-deploy-fixed-wireless-broadband-services-in-developing-countries-07-2004.doc,
2004.

[20] O. W. Purbo. Experiences in deploying a bottom-up ict infrastructure in indonesia. http://sandbox.bellanet.org/ onno/ict4d-roadmap/experiences-in-deploying-a-bottom-up-ict-infrastructure-08-2004.doc,
2004.

[21] A. Sheth, S. Nedevschi, R. Patra, S. Surana, E. Brewer, and L. Subramanian. Packet loss characterization in wifi-based long distance networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 312 –320, may. 2007. `doi:10.1109/INFCOM.2007.44`.

[22] R. W. Stevens. *TCP/IP Illustrated: the protocols*, volume 1. Addison-Wesley, Reading, Massachusetts, USA, 1994.

[23] L. A. Suchman. *Human-machine reconfigurations: Plans and Situated Actions*. Cambridge University Press, Cambridge, UK, 2nd edition, 2007.

[24] G. I. Susman and R. D. Evered. An assessment of the scientific merits of action research. *Administrative Science Quarterly*, 23(4):582–603, Dec. 1978.

[25] W. D. Tucker. *Softbridge: a socially aware framework for communication bridges over digital divides*. PhD thesis, University of Cape Town, Department of Computer Science, 2009. Available from: `http://pubs.cs.uct.ac.za/archive/00000524/`.

[26] W. D. Tucker and E. H. Blake. The role of outcome mapping in developing a rural telemedicine system. In P. Cunningham and M. Cunningham, editors, *IST-Africa 2008 Conference Proceedings*, Windhoek, Namibia, 2008. IIMC International Information Management Corporation. Available from: `http://www.ist-africa.org/Conference2008/outbox/ISTAfrica_Paper_ref_67_doc_2116.pdf`.