Language learning on a next-generation service platform for Africa

Andrew Rice[*] Paula Buttery[†] Idris A. Rai[‡] Alastair Beresford[§]

March 15, 2009

1 Introduction

Developing countries are seeing rapid growth in the deployment of mobile phones. Billions of people (including those in the poorest parts of the world) now have devices capable of communication and computation. In Uganda it was reported that there were 25 mobile phones per 100 people in 2007 [11]. This makes mobile phones the most ubiquitous computing and communication platform in the country compared with computers (1.6 per 100 people), television (2.2 per 100 people) and even radio (15.6 per 100 people). Applications and services for these devices are emerging rapidly.

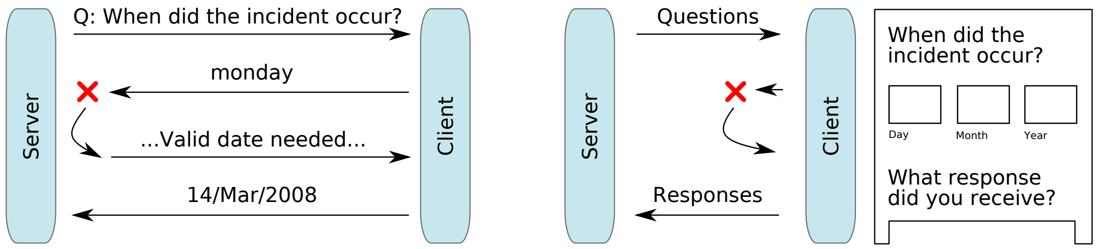
Smart-phones provide a number of interesting properties over traditional handsets. From a hardware viewpoint they provide myriad communications interfaces and large colour screens often with touch-screen capability. Most importantly, however, these devices are increasingly open to 3rd party software development.

These devices are, as yet, not widely deployed even in the markets of developed countries. However, it seems almost certain that they will continue to gain in popularity and eventually achieve high penetration in African markets. We hope that by considering them now we will have the opportunity to help ensure that both devices and software systems evolve appropriately. The open nature of these devices provides a real opportunity to engineer systems which meet the needs of users in developing countries without being hampered by design decisions made for radically different operating environments.

We take this viewpoint based on our work on Computing for the future of the planet [5] in which we are attempting to ask and answer research questions which will improve the positive impact of computing on the larger world.

We intend to investigate the possibilities for a next-generation service platform through the development of a language learning system for mobile devices. In this paper we examine the successes of existing services and applications (Section 2) and motivate our language learning application (Section 4). We go on to outline the capabilities which we are beginning to see in smart-phone devices and relate them to language learning (Section 5) and outline our plans for future work (Section 6).

[*]Computer Laboratory, University of Cambridge, UK. Email: [firstname].[lastname]@cl.cam.ac.uk
[†]Research Centre for English and Applied Linguistics, University of Cambridge, UK
[‡]Faculty of Computing and IT, Makerere University, Uganda
[§]Computer Laboratory, University of Cambridge, UK



(a) Thin-client (b) Thick-client

Figure 1: A thick client implementation of a forms application might reduce network use by validating answers prior to sending and collapsing the response to a questionnaire into a single message.

2 Current applications and service platforms

The simplest services can be used without any additional software installed on the mobile handset. Many of the existing services are deployed using SMS (Simple Messaging Service) messages. Tools such as FrontlineSMS[1] provide compelling functionality in this area by allowing administrators to manage and interact with large contact lists through a simple interface. Numerous applications exist. At one end of the spectrum there are information dissemination services which send notifications and updates to users either in response to specific requests for information or as part of an interest group. At the other end are more interactive services targeting, for example, rural health care. Other mechanisms than SMS are possible. One example is SnapAndGrab which allows users to interact with a digital message board using photo messaging over a Bluetooth connection [7].

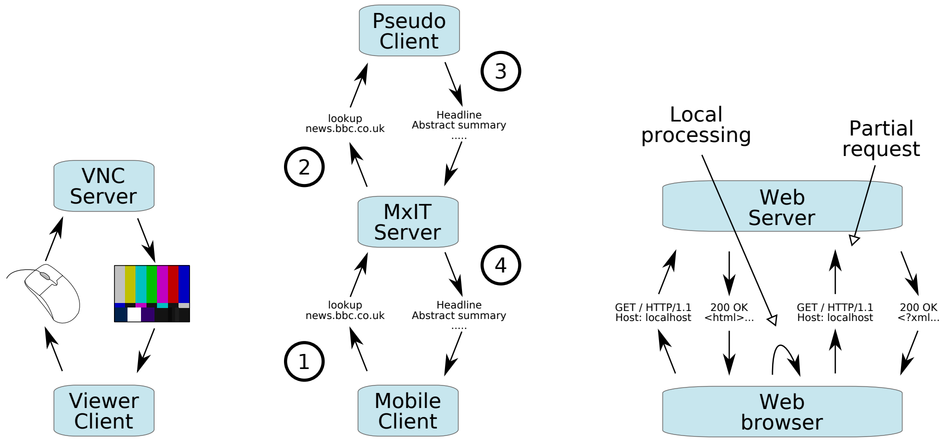
A more complex option is to develop software to run on the handsets themselves. For example solutions such as , FrontlineForms[2] and EpiHandy[3] make use of handset software to interpret data from an SMS message into a questionnaire before sending a response. Mobile banking services such as M-PESA[4] are another hugely popular service. These are superficially integrated into the handset although actions such as revoking a money transfer fall back to either a voice call or an SMS.

Deploying software for handsets incurs a significant one-time cost to users in terms of acquiring and installing a custom application on their phone. One means to mitigate this cost is to provide software which can support numerous applications. The chat program MxIT[5] is a (perhaps unwitting) excellent example of this. By using a general data connection for communication MxIT enables chat room style interaction at a fraction of the cost of an SMS based alternative. This is compelling in its own right and the software has millions of users. However, further services can be deployed through this interface. Math on MxIT is an example which provides mathematics help and advice by allowing students to interact with a tutor [1].

3 Thin-client computing

Virtual Network Computing (VNC) is an early example of a thin-client interface in which applications run on a server machine [10]. Clients connect and use a generic viewer program to interact with these applications. Any changes or additions to the running applications take place on the server—the software on the client is not affected. VNC provides a very simple interface to client

[1] http://www.frontlinesms.com/
[2] http://www.frontlinesms.com/forms/
[3] http://www.epihandy.com
[4] http://www.safaricom.co.ke/index.php?id=747
[5] http://www.mxit.co.za/



(a) VNC (b) MxIT (c) Browser

Figure 2: The various styles of client server interaction cause different patterns of network use.

software by simply copying the graphical output of the server machine to the client and replaying all keyboard and mouse inputs from the client on the server (Figure 2(a)).

It can be argued that MxIT is an example of a text-only thin-client interface. The MxIT client (which is acting as viewer software) simply displays the activity on the server and relays data from the user. New applications (such as Math on MxIT) can be deployed on servers at will as long as they conform to a chat room style of interaction with the user (Figure 2(b)). This is a powerful mechanism for supporting mobile services but has a number of drawbacks.

The most significant problem with a thin-client approach for mobile services is the significant amount of network traffic created. In order to make mobile services as accessible as possible we need to minimize the amount of unnecessary communication. A dedicated (rich/thick) client (such as FrontlineForms) can meet this goal by providing local interaction on the mobile device. This comes at the significant cost of requiring deployment of new software to support new features or different applications.

We believe that a middle-line must be struck between the extremes of thin and thick clients. One approach which has emerged in this space is the current generation of web applications and services which run through a web browser. These provide a mix of online functionality and local processing by executing Javascript (Figure 2(c)). However, extending the web-browser to mobile devices is probably not the best answer. Mobile phones are significantly different devices to desktop machines. In particular peer-to-peer communication is commonly available (through Bluetooth for example) and network communication is often charged pro-rata (whereas a traditional broadband connection is often effectively unlimited use).

We envisage a software platform for smart phone devices which, once installed, allows access to numerous and varied services and applications whilst making effective use of the resources available.

4 Test-case: Language learning on mobile devices

In an attempt to identify useful properties of such a platform we are considering language learning as an example application.

Language learning is an important application in its own right. This is most important in developing countries where there is high ethnolinguistic diversity. In Uganda, for example, there are over

40 living languages [8]. High ethnolinguistic diversity means that English (later joined by Swahili) remained the official language after independence. Today English is spoken by approximately 5% of the population which has a literacy rate of around 50%. Therefore, improving language learning resources has the potential to have a huge impact on improving social mobility and access to government and social services. Similar diversity is present in many other developing countries.

Learning a language requires the cultivation of both interactive and individualised skills over the domains of listening, speaking, reading and writing. Language learning poses interesting possibilities for mobile technology since individualised learning may be undertaken offline while interactive learning can exploit the inherent communicative properties of the device. Additionally, when a device is associated to a unique user it becomes a portal to that learner's language proficiency. Both individualised and interactive applications may then be tailored to be of optimum benefit to the learner. For this to be achieved, it is necessary not only to identify a learners' level of proficiency but also to understand what "level of proficiency" specifically means in terms of language production and comprehension. The English Profile project[6] is a long-term, collaborative programme of research, consultation and publication, designed to enhance the learning, teaching and assessment of English worldwide. The programme aims to produce 'Reference Level Descriptions for English' (i.e. a detailed and objective analysis of what "level of proficiency" means in terms of grammar, vocabulary, and discourse features)[4]. By working in collaboration with projects such as English Profile we hope to fully exploit mobile device technology by designing applications that adapt to accommodate a learners' specific abilities whilst interacting with others learners of differing abilities.

5 Desirable functionality from a next-generation mobile services platform

There are numerous hardware components integrated into modern smart-phones. In this section we review these components and relate them to our goal for a language learning application.

5.1 Communications

A conventional desktop computer has a single, always available, network connection. Mobile phones, however, operate at the other end of the spectrum. They support numerous different connection types and connectivity varies significantly.

The most common types of interface today are Bluetooth, 802.11 (WiFi) and a connection to the mobile network (for example GSM or UMTS). Bluetooth connections consume the least power and are designed for low bandwidth, short range communication either between devices or with peripherals. WiFi connections cover local area wireless infrastructure networks but also support ad-hoc connections between hosts. These connections provide an order of magnitude more bandwidth than the other interfaces but at the cost of significant power drain. The connection to the mobile network is usually a lower bandwidth, always on connection used for receiving and making calls or for data access when other options are unavailable.

Most significantly, the Bluetooth and WiFi interfaces support local communication without involving the network operator and so have huge potential for saving network bandwidth and hence lowering costs to the user.

Language learning (particularly in a school environment) often entails groups of students working on similar content. These peer-to-peer interfaces could allow students to share content once it has been retrieved from the network and to facilitate peer learning through cheap, effective

[6] http://www.englishprofile.org/

communication. Supporting this data-transfer can either be done with local infrastructure such as WiFi hotspots or in a totally ad-hoc manner based on natural interactions between learners. The research areas of Mobile Ad-hoc Networks (MANETS) [2] and Delay Tolerant Networks (DTNs) are highly relevant here.

Continually changing connectivity causes a problem for thin-client systems which are not designed for disconnected operation. However, recent advances have been made by vendors in an attempt to support offline operation of web-applications. One example of this is Google Gears[7] which provides (among other features) local data storage for applications. Currently this is designed to support either connected or disconnected modes of operation. For a mobile device we must also consider the availability of connections to peers.

Assessment is another vital aspect of language learning because it provides feedback to the learner as to their progress and allows teachers and providers to tune their content. This kind of data is interesting because it is delay tolerant—it is harvested in the background and there is a large amount of leeway in when the data must be uploaded. This would be beneficial if the service framework waited to send the data until a cheaper network connection (perhaps from a WiFi hotspot) was available.

Another possibility is to provide infrastructure to allow casual unstructured working in the same style as the Amazon mechanical turk[8]. This system automatically distributes jobs to a pool of human workers and collects the results. Maintaining a connection to the mobile network (in order to receive calls) has no financial penalty and so a company could recruit workers to provide interaction or assessment and spontaneously route incoming requests from learners to active workers. The option to connect foreign learners of African languages with native speakers could develop an interesting revenue stream into the continent.

5.2 Input mechanisms

One outcome of the Math on MxIT project was the observation that the chat room approach is easily transferable to learning in other subjects—with the exception of language learning [1]. This is because the limited input capabilities of a phone give rise to creative spelling and grammar when text messaging. This is mitigated to a small extent by the keyboards which smart phones increasingly incorporate in their designs. However, other innovations such as touch-screen interfaces and (more rarely) a stylus provide opportunity for other mechanisms of interaction. Fast and accurate data entry remains a long way off however and so it remains to be seen which aspects of language learning (or literacy) can be taught effectively.

Other input devices such as the microphone and camera can provide further interaction methods although at the cost of expensive computation (for machine vision or audio recognition) or increased network use (for off-loading processing onto a server).

Many phones now incorporate a GPS device which can produce location information. This can be used to provide context based learning either by responding to the co-location of peer learners or by providing context specific teaching. One example of this might be to teach domain specific vocabulary when a user enters a particular region.

5.3 Output mechanisms

Phone manufacturers have been steadily increasing the resolution, colour capability and size of displays. In addition to this the graphical capability of the hardware has continued to increase and 3D graphics or video playback are now feasible (if power-intensive) on many handsets. Mobile

[7]http://gears.google.com/
[8]https://www.mturk.com/mturk/welcome

phones have also acquired many of the capabilities of portable music players. This provides the possibility for providing varied and interesting content to learners through graphical and audio content.

Despite the mitigating effects of peer-to-peer communication it might not be appropriate to distribute full video content due to the communications (and storage) costs inherent. As an alternative, generating images and video on the phone presents a number of interesting questions regarding data coding and compression.

5.4 Onboard computation

A thin-client solution leaves almost all computation on the server at the expense of increased communication costs. A thick-client solution runs all computation on the device (with associated increased power consumption) and can therefore minimise network costs.

Ideally a service platform should allow trade-offs between these two extremes. This is an example of task allocation in a network [3] in which one must consider the tradeoffs between local and remote computation. Software development and deployment is most likely to be quicker for server-side computation. However, local computation will provide faster responses to the user for interactive tasks and can act to minimise network consumption. This is a dynamic problem. For the periods that the device has cheap network access, server-side execution might be desirable and for periods using a paid network a user might prefer to switch to client-side execution.

Some examples for language learning might be locally grading and assessing a learner and only communicating the result if new content is required—this is beneficial for the privacy of the learner too. Alternatively, lexically based (e.g. flash card) learning applications would benefit from local execution due to the high cost and latency of continually relaying the content and the response over the network.

Another benefit of client-side computation is reduced reliance on server infrastructure. There are large energy and environmental costs to providing providing high availability services in datacentres: in part because of application demands that these systems should be always available. Relaxing this constraint by permitting more client-side computation would allow significant power reductions in the datacentre [9].

6 Future work

We are beginning by developing prototype learning applications for Google Android[9] handsets and aim to run a test-program based at Makerere University, Kampala. We hope this collaboration will help to ease the particular challenges which arise in designing technology for developing countries [6]. We are interested in finding out about other services, related projects and any other issues of importance which we have overlooked.

7 Conclusion

Smart-phone devices provide notably more features and flexibility than traditional handsets and are seeing rapid adoption. It is only a matter of time before these devices begin to penetrate the African market. We are interested in how one might engineer a software platform to allow these devices to efficiently access services and information whilst exploiting the potential benefits of their increased communications, storage and computation ability. We hope to inform our thoughts and development of this platform by constructing a number of applications and services starting with a language learning application which we aim to test in Uganda.

[9] http://www.android.com/

8 Acknowledgements

Many thanks are due to Andy Hopper for his work with us on Computing for the future of the planet and to Gary Marsden for engaging with us and sharing his experiences and research.

References

[1] L Butgeriet. Math on MXit: using MXit as a medium for mathematics education. In *Meraka INNOVATE Conference for Educators*, CSIR, Pretoria, April 2007.

[2] M. Conti and S. Giordano. Multihop ad hoc networking: The theory. *Communications Magazine, IEEE*, 45(4):78–86, 2007.

[3] Jonathan J. Davies, Alastair R. Beresford, and Alan Mycroft. Language-Based optimisation of Sensor-Driven distributed computing applications. In *Fundamental Approaches to Software Engineering (FASE)*, pages 407—422, April 2008.

[4] J.A. Hawkins and P. Buttery. Using learner language from corpora to profile levels of proficiency: Insights from the english profile programme. In *Studies in Language Testing: The Social and Educational Impact of Language Assessment*.

[5] Andy Hopper and Andrew Rice. Computing for the future of the planet. *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, 366(1881):3685–97, October 2008. PMID: 18672457.

[6] Matt Jones and Gary Marsden. *Mobile Interaction Design*. Wiley, 2006.

[7] Andrew J. Maunder, Gary Marsden, and Richard Harper. SnapAndGrab: accessing and sharing contextual multi-media content using bluetooth enabled camera phones and large situated displays. In *CHI '08 extended abstracts on Human factors in computing systems*, pages 2319–2324, Florence, Italy, 2008. ACM.

[8] G Raymond, editor. *Ethnologue: Languages of the World, Fifteenth edition*. SIL International, 2005.

[9] Andrew Rice, Sherif Akoush, and Andy Hopper. Failure is an option. Technical Report MSR-TR-2008-61, Microsoft Research Technical Report.

[10] T. Richardson, Q. Stafford-Fraser, K.R. Wood, and A. Hopper. Virtual network computing. *Internet Computing, IEEE*, 2(1):33–38, 1998.

[11] International Telecommunications Union. World Telecommunication/ICT indicators database 2008, 12th edition. Technical report.