# ICT4EGRA: A Technology System for Early Grade Reading Assessment Data Collection

Max Heinritz

University of Pennsylvania

Spring 2011

Advisors: Professor Wagner, Professor Ungar, Katie Murphy

## Abstract

This paper introduces a fully functional ICT system for EGRA data collection. The Early Grade Reading Assessment (EGRA) is an education evaluation instrument used to measure student literacy in developing countries. The test is administered by trained enumerators who assess samples of students at multiple schools. Results are currently recorded on paper and then manually digitized when enumerators return from fieldwork. With the ICT system introduced here, results would instead be recorded on mobile phones and transmitted over the Internet. The system relies heavily on Open Data Kit, a set of open-source software for mobile data collection, storage, and aggregation. The results are analyzed online with Google Apps Script.

## 1. Introduction

The Early Grade Reading Assessment (EGRA) is an education evaluation instrument used to measure student literacy in developing countries. It focuses on the most basic testable reading skills such as letter name knowledge, listening comprehension, and word reading (RTI International, 2009). Since its inception in 2006, the assessment has been deployed in more than 40 countries and translated into dozens of languages (RTI International, 2010).

EGRA is typically administered by independent evaluators rather than local teachers or school administrators. The evaluators work in teams, travelling from school to school, administering the assessment to a sample of students in each grade (RTI International, 2009, p. 40). The results are recorded on paper as the test is administered and then manually entered into a computer when the teams return from the field (RTI International, 2009, p. 43).

Pouezevara and Strigel (2011) suggest that information and communication technologies (ICT) could improve both the efficiency and accuracy of EGRA data collection. This project heeds their findings and introduces a fully functional system for EGRA data collection using Open Data Kit (ODK). The paper is divided into five main sections: motivation, an overview of EGRA, the technical proposal, discussion, and future directions.

## 2. Motivation

This project is motivated by a number of social and economic trends in the developing world: the desire for universal education, the rapid adoption of ICT, poor evaluation of ICT-for-education (ICT4E) projects, and lessons from ICT data collection and service delivery systems in other sectors.

### A. Universal Education

Education is a priority of the international development community. In 1990, 155 countries and 150 organizations pledged their support for the Education for All initiative, a formal collaboration among countries, non-governmental organizations, academics, and others with the goal of education for all by the year 2000 (UNESCO, 2011). International support for education reform was renewed in 2000 with the Millennium Development Goals (MDGs), eight widely adopted targets for poverty reduction set forth by the United Nations to be completed by 2015. One of these goals seeks universal education: to "[e]nsure that all boys and girls complete a full course of primary schooling," where "primary school" refers to the first five to eight years of formal education (UNDP, 2011). Progress on this goal has been measured with three data points:

1. Net primary school enrolment ratio; that is, the ratio of children of primary school age that are currently enrolled in school (UNESCO Institute for Statistics, 2004)
2. Proportion of pupils starting primary school who reach the last grade of primary
3. Literacy rate of adults aged 15 to 24

These data points are only proxies for the underlying target of universal education, and, as with most measurement systems, their structure affects the incentives of the organizations and countries attempting to achieve the underlying goal. In particular, low- and middle-income countries have increased their enrolment figures, with world net primary school enrolment now approaching 90% (UNESCO Institute for Statistics, 2011), while appearing to neglect the learning that goes on once the students are in the classroom (World Bank: Independent Evaluation Group, 2006). The trend is clear in the data. The OECD's Program for International Student Assessment (2004) reports that *median* learners in low income countries are still only comparable to the *third* percentile of students in developed countries.

The World Bank's Independent Evaluation Group (2006, Introduction, p. 6) expressed concern that, unlike other global education initiatives such as Education for All, "*the MDGs primarily address the issue of access to primary education and do not include an explicit goal with respect to either the quality of instruction or to learning outcomes*" (emphasis in original text). Part of the reason why the MDGs explicitly target access but not actual learning outcomes may be because education quality is more difficult to measure than basic enrolment (Wagner, 2011).

Until 2006, many national and international standardized tests were administered only to students in grades four and above and assumed preexisting basic levels of literacy and numeracy (RTI International, 2009). Students often perform so poorly on these national, standardized tests that education evaluators are unable to use the results to inform education policy or encourage accountability (RTI International, 2009). If everyone fails an exam, it is unclear what needs to change or who is responsible.

In response to these concerns, the United States Agency for International Development (USAID) contracted RTI International, a nonprofit research institute, to build a low-cost, early-grade literacy test for use in developing countries. RTI consulted cognitive scientists, education ministries, teachers, and assessment experts and produced the first sample EGRA instrument in October 2006 (RTI International, 2009).

**B. Rapid Adoption of ICT**

Another socioeconomic trend in the developing world is growth of modern ICT. The World Bank (2002, p. 3) defines ICT as "hardware, software, networks, and media for collection, storage, processing, transmission, and presentation of information." Use of these ICTs has grown dramatically in the last two decades (World Bank, 2010). Driven primarily by wireless technologies and telecommunications deregulation, mobile phone use in developing countries now represents more than half of the 4.5 billion mobile subscriptions worldwide (World Bank, 2010). The number of Internet users has also grown dramatically and is now approaching 1.8 billion, with almost 900 million users in low- and middle-income countries (World Bank, 2010). Incredibly, more people today have access to a mobile device than to a toilet (World Bank, 2010; WHO & UNICEF, 2008).

The motivation for ICT-based data collection systems is clear. With cheaper hardware and better connectivity, ICT data collection systems are now cost-effective alternatives to their paper-based counterparts, even in low-income countries (Pouezevara & Strigel, 2011).

**C. Poor Monitoring and Evaluation of ICT-for-Education (ICT4E) Projects**

The trends described above—the desire for universal education and the adoption of ICT in developing countries—have inspired dozens of ICT projects targeting access to education and learning outcomes. One of the longest-running ICT4E projects is Interactive Radio Instruction (IRI), an instructional tool originally developed by Stanford University in 1974 to deliver "active learning by radio" (Ho & Thukral, 2009). USAID has funded many IRI implementations the decades since, and these projects' well-funded evaluation components have repeatedly proven the system's effectiveness (Tilson, et al, 1991; Leigh, 1995; Bosch, 1997; Ho & Thukral, 2009).

However, due in part to its uncommonly generous funding from USAID, IRI is exceptional among ICT4E projects in the strength of its monitoring and evaluation programs (Ho & Thukral, 2009; Trucano, 2010). Michael Trucano, Senior ICT & Education Specialist at the World Bank, has expressed concern about the "still-weak evidence base we have demonstrating positive, cost-effective uses of other [non-IRI] ICTs in educational settings in [developing countries]" (Trucano, 2010). Other researchers and practitioners share his sentiments (Wagner, 2005; Ford & Botha, 2010). The challenge for ICT4E projects—and development projects more generally—is that evaluations are often considered low-priority by donors and therefore are either poorly funded or not performed at all.

A widely used ICT system for EGRA data collection could enable better monitoring and evaluation in the ICT4E space by allowing development practitioners to more cheaply evaluate their work. For example, when a USAID-funded education ministry builds an EGRA instrument for the ICT system, it could then make the code for instrument freely available online to NGOs and researchers operating in their country. These practitioners could evaluate ICT4E projects aimed at early grade reading using the same EGRA instrument. Similarly, such an ICT system could be used for other education assessment beyond existing EGRA instruments.

### D. ICT Systems in Other Sectors

This project is also motivated by past work in ICT for development. Digital data collection is common in standard survey research, and increasingly common in development work. Past projects have built ICT systems for data collection and service provision for health (Luk, Ho & Aoki, 2008), agriculture (Patel, Chittamuru, Jain, Dave & Parikh, 2010), microfinance (Parikh, 2006), and other sectors. The literature emphasizes importance of the social embeddedness of computing, and the challenges of bringing new technology to the developing world. The lessons of past ICT projects motivates the relatively small scope of this one: here we focus on a specific need of a pre-existing, well-embedded development project.

## 2. Early Grade Reading Assessment (EGRA)

These broad social trends motivated a project involving ICT, education, and monitoring and evaluation. The EGRA instrument was chosen in particular because it is widely used (RTI

International, 2009), and because there is a clear need for an ICT data collection system (Pouezevara & Strigel, 2011).

Since its pilot launch in 2006, EGRA has been deployed across the world by donor agencies and education ministries (RTI International, 2010). The largest assessment to date took place in Ethiopia in summer 2010, featuring six languages, two scripts, and 9,200 students (USAID, 2010).

EGRA appears to be both valuable and sustainable. Though most deployments so far have been funded by aid agencies rather than the countries themselves, their implementations occurred in direct collaboration with education ministries and other local stakeholders (RTI International, 2010). Early adopters of EGRA such as Guyana, Honduras, Nicaragua, and Haiti have used their EGRA results to adjust education policy and, after observing the success of their pilot programs, have elected to either continue EGRA evaluation or incorporate EGRA-style assessment techniques into their national tests (Garant, 2010).

**Current EGRA Data Collection Techniques**

EGRA is a one-on-one assessment in which each evaluator tests exactly one student at a time. It comprises many components, or subtests, designed to measure specific reading skills (RTI International, 2009). Different implementations of EGRA may use different components depending on the local languages and administration resources. The sample EGRA assessment designed by USAID and RTI International (2009) contains eight components, described further in Appendix A. The whole test takes 15- to 20-minutes per student.

An example component is “Letter Name Knowledge,” which measures letter recognition skills by having students read aloud the names of upper- and lower-case letters shown in random order on a printed piece of paper in the student pack (RTI International, 2009). Many EGRA components are timed and “timed-limited,” meaning that after a certain point the evaluator stops the components regardless of completion (RTI International, 2009).

For each component, there is an administrator sheet with instructions on how to administer the exercise and a place for the evaluator to record the test results. A sample instructor sheet for the “Letter Name Knowledge” component is shown in Appendix B (RTI International, 2009). This sheet contains a list of all the letters the child is to read as well as word-for-word instructions to be read aloud to the student. As the student attempts to identify the letters on the student sheet, the evaluator marks on the evaluator sheet which letters the student fails to correctly identify. The total time taken is also recorded at the end of the exercise. Each evaluator uses a new sheet for each student and can test between 12 and 15 students in one school morning (RTI International, 2009, p. 62).

In addition to the actual individual test results, the EGRA teams also collect school-level information (number of students, location, contact information, etc), student selection details (which students were chosen to be tested and how), and responses to optional parent and student surveys (Pouezevara & Strigel, 2011). The optional parent and student surveys are not part of the core EGRA assessment, but their questions help contextualize EGRA results. Questions include “do you have a textbook?” and “does someone at home help you with your homework?” (Pouezevara & Strigel, 2011, p. 3).

When the team completes the assessment, papers with data gathered in envelopes and transported to data entry centers in boxes. Trained local data-entry specialists read the responses from the sheets and manually enter them into a computer system for statistical analysis with software packages such as Excel or Stata (RTI International, 2009).

## 3. ICT4EGRA: A System for Digital EGRA Data Collection

The author set up a fully functional system (nicknamed "ICT4EGRA") for collecting EGRA results digitally using Open Data Kit. With ICT4EGRA, the evaluator records results on a mobile phone instead of the evaluator sheet. The student sheets are not replaced. Results are saved on the mobile phone while the instrument is being administered and can later be uploaded to a server. The server software aggregates results from multiple phones and compiles it for further analysis. An ICT4EGRA system has been implemented for demonstration purpose, but like the EGRA instrument itself, the method is general and could be deployed again.

**Open Data Kit**

The core of the system is Open Data Kit (ODK), "a free and open-source set of tools which help organizations author, field, and manage mobile data collection solutions" (Open Data Kit, 2011). ODK is developed in part by computer science students at the University of Washington and funded in part by Google.org (Hartung et al, 2009). The system is robust, actively developed, and deployed in hundreds of research projects around the world, including some developing regions without universal connectivity (Hartung et al, 2009). It provides out-of-the-box solutions for most of the technical needs of the ICT4EGRA system.

ODK comprises a number of pieces of software: ODK Collect is an Android application for data collection in the field; ODK Aggregate is "click-to-deploy" server software which processes data uploaded from ODK Collect; ODK Build is a graphical drag-and-drop web interface for building custom JavaRosa XForms surveys for ODK Collect and ODK Aggregate (Open Data Kit, 2011). XForms is a standard file format for data input and form design, and the JavaRosa specification is a subset of the XForms format which ODK Collect renders properly.

**Deploying ODK for EGRA Data Collection**

The process of deployment involves the following: 1) configuring software, 2) building an XForms file for each component, 3) combining the XForms files for each component into larger multi-component files, 4) downloading the XForms files on to phones, 5) administering the test, 6) sending the results from the phone to the server, and 7) analyzing the aggregated results. Further instructions are available at:

`http://www.seas.upenn.edu/~heinritz/ict4egra/`

**Step 1: Configure software**. ODK Aggregate can be installed on to Google App Engine, a free web hosting platform which allows Google Account holders to quickly deploy scalable web applications. Each ICT4EGRA deployment will need its own instance of the server software running to store and manage data. In demonstration system built for the sample EGRA instrument in the *EGRA Toolkit* (2009), ODK Aggregate runs at

`http://ict4egra.appspot.com`

ODK Collect can be configured to communicate directly with a server running ODK Aggregate. The Android application can download new XForms files from the server and send back captured data. The author ran an Android virtual device to quickly test the system. Further instructions on setting up packaged software can be found in Appendix C.

**Step 2: Build an XForms file for each EGRA component**. A complete EGRA instrument could be built and tested in a single XForms file, but the author found it easiest to build and test files for each EGRA component separately and then later combine the individual components into a single file. Appendix D provides the XML source code for Letter Name Knowledge, the first component of the sample EGRA instrument provided by RTI International in the *EGRA Toolkit* (2009).

For simple EGRA components, ODK Build can be used to generate the necessary XML. For more complex forms, manual form construction may be necessary. While implementing

ICT4EGRA for the sample EGRA instrument in the *EGRA Toolkit* (RTI International, 2009), the author used several scripts to generate XML code. Appendix E provides a sample script that updates the letters in the Letter Name Knowledge XML file, taking as input a file containing 100 letters separated by spaces and the XML source code in Appendix D. Such scripts will hopefully allow the same XML source code to be used in multiple ICT4EGRA deployments.

XForms files can be validated using ODK Validate and tested locally on an Android virtual device. Appendix E provides screenshots of ODK Collect rendering the survey source code for component one (letter name knowledge). The author built individual XForms files for each of the eight components of the sample EGRA instrument in the *EGRA Toolkit* (2009).

The final component, dictation, requires the student to listen to a series of words spoken aloud by the evaluator and write down what he or she hears. Since the results for this component are recorded on the student sheet rather than in the evaluator sheet, the XForms file prompts the evaluator to take a picture of the student response form using the mobile phone's camera. The picture can be uploaded to the server with the rest of the EGRA results and analyzed after the teams return from the field, saving valuable data entry time at the schools.

In addition to the eight EGRA components, the author also built a short XForms form which prompts the evaluator to enter the student's ID number (if any), last name, first name, gender, and grade level. This information will link EGRA results to individual students and hopefully allow teachers and evaluators to better track and improve individual-level learning outcomes.

**Step 3: Combine the files for each component into a larger survey**. The shell script provided in Appendix G automates the combination process: it takes as input an EGRA survey template file (show in Appendix H) and multiple component XForms files. It then adds the code from the component XForms files to the template file to create a single XForms file with all components. This script allows users to pick and choose which components to include in the EGRA instrument.

The author built two final XForms files: `EGRA_C1toC7.xml` (with components 1 to 7) and `EGRA_C8.xml` (with component 8). Component eight requires the evaluator to take a picture using the mobile phone, and the camera software on Android is buggy. In testing, component eight would cause ODK Collect to crash and lose the data collected from the first seven components. To isolate the effects of this type of crash, components one to seven are administered separately from component eight.

**Step 4: Send the XForms file to ODK Aggregate and ODK Collect**. Once the XForms file is built, ODK will take of most of the technical work of capturing, storing, aggregating, and displaying the data. To enter the XForms file into the ODK system, upload the final .xml file to ODK Aggregate using its web interface. Then, from an ODK Collect instance on an Android device, select “Manage Forms And Data” and “Get New Forms.” ODK Collect will find the XForms file on ODK Aggregate and download it to the phone.

**Step 5: Administer the assessment**. Many phones can capture data for the same XForms form, so many evaluators can use the system simultaneously. Results are stored locally on the phone in an XML after the test is administered. Each student’s results are saved on the phone’s local memory by ODK Collect as separate results files.

**Step 6: Send results from phones to the server**. Results can be sent from the phone to the server either over a 3G mobile connection or through a wireless LAN connection. This is an out-of-the-box ODK Collect feature. Results could also be copied from the phone to a computer and then uploaded to the server through the ODK Aggregate web interface.

**Step 7: Download and analyze the results**. The ODK Aggregate web interface allows users to view gathered data online and download results in a comma-separated values (.csv) file. This file format is standard so the results can be analyzed using any common statistics software. For demonstration, the author wrote a Google App Script function (see Appendix J) for Google Spreadsheets to parse raw data, compute scores for each component, and calculate averages across students for each of the scores.

## 4. Discussion

ICT4EGRA is fully functional from a technical perspective, and we expect benefits to exceed costs from a policy perspective as well. We do not know of any other system like ICT4EGRA that has been deployed yet. The for-profit education company Wireless Generation offers a commercial product with similar features called mCLASS, but mCLASS is based on an alternative to EGRA called DIBELS (for Dynamic Indicators of Basic Early Literacy Skills) (Wireless Generation, 2011). DIBELS is widely used in the United States and targets a slightly different use case than EGRA. mCLASS software can be run on mobile devices and tracks learners, schools, and districts over time and grade levels. The software is robust and expensive, whereas ICT4EGRA is light and cheap.

### A. Benefits of ICT in EGRA Data Collection

The ICT4EGRA system could improve existing data collection techniques by making them more time and cost efficient. It would help to reduce the amount of paper used, to reduce the time needed for data entry, and to improve data quality (Pouezevara & Strigel, 2011).

In an EGRA deployment in Mali, some 156,000 pieces of paper were needed to evaluate 10,200 children (Pouezevara & Strigel, 2011). In addition to printing and paper costs, non-digital data collection adds significant stapling, handling, and organizational overhead. This overhead was estimated to be 15 percent of the overall budget for a recent deployment in Liberia (Pouezevara & Strigel, 2011). Further, some schools are reachable only by motorcycle, and papers can be lost or damaged in transit (Pouezevara & Strigel, 2011).

A digital data collection system would save time by removing the step of manual data entry. It currently takes a trained data entry specialist ten minutes to digitize the results from one assessment form (Pouezevara & Strigel, 2011). The time saved by digital data collection means that results could be available for schools and evaluators in days rather than weeks after the assessment.

The accuracy of data entry will also improve by skipping the manual data entry because one source of human error is removed. In a recent deployment, results from 16 of 50 schools were digitized incorrectly because of human error in data entry (Pouezevara & Strigel, 2011).

**B. Challenges of ICT for EGRA Data Collection**

The system also introduces new challenges. The system stores data locally on phones that could be lost or damage. The risk of losing data could be mitigated by sending evaluator teams into the field with a laptop and a USB cord so that they could copy the results files from the phones to the laptop at night. Similarly, if the phones are being used in a place with a 3G connection, results could be uploaded to the server instantly.

ICT4EGRA may also introduce new sources of human error, especially if the interface is confusing to enumerators. In the Letter Name Knowledge component, the evaluator is asked to mark incorrectly identified letters on the evaluator sheet. For the assessment with ODK we reversed the practice: a check indicates correctness. We chose to do this because on Android a green check mark is the system icon for a check box (see screenshots in Appendix F). It may be difficult for evaluators to mark each letter if the student is reading them quickly and correctly. It may also be awkward for the evaluator to scroll down a list while the student is reading them aloud. Fortunately, the enumerators are already trained as a part of EGRA so it will be easy for them to be trained to use the phones.

Another challenge is cost. We expect that the costs of mobile phones and laptops will be less than paper and data entry, but these figures will vary from project to project (Pouezevara & Strigel, 2011). Digital data collection method may also reduce the ability of enumerators to record qualitative information by writing comments in the margins of the evaluator sheets (Pouezevara & Strigel, 2011).

## 5. Future

The obvious next step in the project is consultation with professional EGRA implementers. Their feedback will help decide if a pilot deployment makes sense. With their support, we hope that the system is tested in the field and developed further.

Fortunately ICT4EGRA uses very little custom-build software; it relies heavily on free software that is easy to set up. Most of the code is maintained and documented by third parties, so it will be relatively easy for EGRA implementers to deploy the software without dedicated technical talent. The entire ICT4EGRA system described here can be replicated in a few days (or hours) by someone with limited technical experience.

In addition to a pilot deployment, we see two future directions for work on the project. The first is to implement new features to make the system more useful. The second is to explore in theory how ICT could change education evaluation instruments in the developing world.

### A. New Features

Most features of ICT4EGRA are out-of-the box features of ODK. More features would make the system more robust. Here are some desirable features:

**Replace the timer**. While administering the assessment, EGRA evaluators often have difficulty managing the student, the response forms, and the timer (Pouezevara & Strigel, 2011). The existing ICT4EGRA system does not replace the timer, but future systems should. The source code for the Android application will need to be modified.

**Support multiple languages**. The ICT4EGRA system is currently coded only for English, but ODK JavaRosa XForms provide out-of-the-box support for translation. With the help of a translator, the system could be easily translated to other languages. The character encodings are limited to those supported on Android.

**Offline data analysis**. The Google Spreadsheet method for processing the raw data requires an Internet connection. However, because data is stored on locally on the phone immediately after the assessment is given, an evaluator with a computer and a phone-to-computer USB cord could potentially retrieve data from the phone and analyze it locally on her computer while in the field. This would allow an EGRA team to capture data on phones, copy the results to a laptop, process the data, and inform the school teachers and parents of the test scores in the same day. Real-time feedback will make EGRA results more meaningful to students and schools tested and better inform education policy decisions.

**Instant feedback**. Similarly, in areas with high Internet connectivity, the system could be configured to process data from the field as soon as the server receives it. Same-day results could be sent to teachers and other stakeholders through mobile phones or email.

**B. Long-term Potential of ICT for EGRA**.

This project will hopefully inform future attempts to digitally collect education assessment data. We identify three theoretical ways that the system could be developed:

**New EGRA components**. ICT could enable testers to easily capture different types results from new components such as the Peabody Picture Vocabulary test. ICT could also be used to dynamically update the components as they are administered so that a student receives a more difficult question if he answers a question correctly and an easier question if as he answers incorrectly. This style is similar the GMAT and may help better pinpoint the learner's actual literacy level. This functionality would be difficult to implement without ICT.

**New EGRA testing format**. Technology could potentially change the way EGRA is administered entirely. The Pico projector is a relatively cheap ($300) phone-sized portable computer project that could be used to administer assessments to groups of students at once. Other new technologies like portable scanners, written character recognition, and digital pens could be used to change how students take EGRA-like tests (Pouezevara & Strigel, 2011).

**ODK to measure other educational outcomes**. Open Data Kit surveys could be used to capture more that just EGRA results. Data from other education evaluation instruments could be collected using similar systems. These systems should be standardized and made available online for independent evaluators to measure the results of their own education interventions. Such a standard may improve the poor standards for monitoring and evaluation of ICT for education projects.

## 6. Conclusion

This paper introduced a fully functional ICT system for EGRA data collection. It described how such a system could be deployed in the field and proposed directions for future work. The project is motivated by a number of high-level social and economic trends including the UN MDG for universal education and the rapid adoption of ICT by low- and middle-income countries. The proposed system relies heavily on existing software packages including Android, Open Data Kit, and Google Spreadsheets. The system is fully functional from a technical perspective but has not been tested in an actual EGRA implementation. The author hopes that EGRA implementers find this project compelling enough to warrant a pilot assessment in the field.

References

Bosch, A. Interactive Radio Instruction: Twenty-Three Years of Improving Education Quality. Education and Technology Notes, Vol. 1, No. 1, World Bank, Washington DC, 1997.

Ford, M. & Botha, A. (2010) A Pragmatic Framework for Integrating ICT into Education in South Africa. IST-Africa 2010 Conference Proceedings.

Garant, M. (2010). Post-EGRA: What actions have countries taken. *USAID Education Data for Decision Making News*. Retrieved April 12, 2011 from https://www.eddataglobal.org/news/index.cfm?id=32&fuseaction=detail

Google. (2011). Uploading and exporting: Importing. *Google Docs Help*. Retrieved April, 2011 from http://docs.google.com/support/bin/answer.py?hl=en&answer=40608&topic=15171

Hartung, C., Anokwa, Y., Brunette, W., Lerer, A., Tseng, C. & Borriello, G. (2009). Open Data Kit: Tools to Build Information Services for Developing Regions. Google: Mountain View, US.

Ho, J. & Thukral, H. (2009). *Tuned in to student success: Assessing the impace of interactive radio instruction for the hardest-to-reach*. Washington DC: Education Development Center.

Leigh, S. Changing Times in South Africa: Remodeling Interactive Learning. LearnTech Case Study Series #8, USAID, Washington DC, 1995.

Luk, R., Ho, M. & Aoki, P. (2008). Asynchronous Remote Medical Consultation for Ghana. In *Proceeding of the Twenty-Sixth Annual ACM Conference on Human Factors in Computing Systems* (CHI '08), p. 743-752, Florence, Italy.

Open Data Kit. (2011). Magnifying Human Resources Through Technology. Retrieved April, 2011 from http://opendatakit.org/

Organization for Economic Co-operation and Development (OECD). (2004). *Learning for tomorrow's world: First results from PISA 2003*. Retrieved April 12, 2011 from http://bit.ly/ijSw1h

Parikh, T. (2006). Rural Microfinance Service Delivery: Gaps, Inefficiencies and Emerging Solutions. *IEEE/ACM International Conference on Information and Communication Technologies and Development*. Berkeley, CA.

Patel, N., Chittamuru, D., Jain, A., Dave, P. & Parikh, T. (2010). Avaaj Otalo — A Field Study of an Interactive Voice Forum for Small Farmers in Rural India. *CHI 2010: HCI and India*. Atlanta, GA.

Pouezevara, S. & Strigel, C. (2011). Using Information and Communication Technologies to Support EGRA. Draft. Paper commissioned by RTI International.

Research Triangle Institute International (RTI International). (2009). *Early Grade Reading Assessment Toolkit*. Retrieved April 12, 2011 from https://www.eddataglobal.org/documents/index.cfm?fuseaction=pubDetail&ID=149

RTI International. (2010). *Early Reading Assessment Country Tracker*. Retrieved April 12, 2011 from https://www.eddataglobal.org/documents/index.cfm?fuseaction=pubDetail&ID=188

United States Agency for International Development (USAID). (2010). Largest EGRA Application takes place in Ethiopia. *USAID Education Data for Decision Making News*. Retrieved April 12, 2011 from https://www.eddataglobal.org/news/index.cfm?id=21

United Nations Development Program (UNDP). (2011). Goal 2: Achieve universal primary education. *United Nations Millennium Development Goals*. Retrieved April 12, 2011 from http://www.undp.org/mdg/goal2.shtml

United Nations Educational, Scientific and Cultural Organization (UNESCO). (2011). Education for all. *UNESCO Education*. Retrieved April 13, 2011 from http://www.unesco.org/education/efa/ed_for_all/

UNESCO Institute for Statistics. (2004). *Global Education Digest 2004*. Montreal: Author.

UNESCO Institute for Statistics. (2011). Enrolment ratios by ISCED level. *Global Education Database*. Retrieved April, 2011 from http://stats.uis.unesco.org/unesco/ReportFolders/ReportFolders.aspx

Tilson, T., Jamison, D.T., Fryer, M., Godoy-Kain, P., Imhoof, M. The Cost Effectiveness of Interactive Radio Instruction for Improving Primary School Instruction in Honduras, Bolivia and Lesotho. Paper presented at the CIES Annual Conference in Pittsburgh, Pennsylvania, on March 15, 1991.

Trucano, M. (2010). Interactive Radio Instruction: A Successful Permanent Pilot Project? *EduTech: A World Bank Blog on ICT use in Education*. Retrieved April, 2011 from http://blogs.worldbank.org/edutech/iri

Wagner, D. A. (2005). Monitoring and Evaluation of ICT4E: An Introduction. *Monitoring and Evaluation of ICT in Education Projects: A Handbook for Developing Countries*. Washington DC: World Bank Group.

Wagner, D. A. (2011). Smaller, quicker, cheaper: Improving learning assessments for developing countries. Draft. Paper commissioned by the IIEP-UNESCO and the Fast Track Initiative. Philadelphia: International Literacy Institute.

Wireless Generation. (2011) mClass: DIBELS Next. Retrieved April, 2011 from http://www.wirelessgeneration.com/assessment/mclass-dibels/overview

World Bank: Development Data Group & Global ICT Department. (2010). The Little Data Book on Information and Communication Technology 2010. Washington, DC: World Bank Group.

World Bank: Global ICT Department. (2002). Information & Communication Technology Sector Strategy Paper. Washington, DC: World Bank Group.

World Bank: Independent Evaluation Group. (2006). *From schooling access to learning outcomes—An unfinished agenda: An evaluation of World Bank support to primary education*. Washington, DC: World Bank Group.

World Health Organization (WHO) & United Nations Children's Fund (UNICEF). (2008). *Progress on Drinking Water and Sanitation*. United States: Author.

# Appendix A

## Sample EGRA Components

This table is taken from "Exhibit 8. Review of Instrument Components" in the RTI International *EGRA Toolkit* (2009, p. 21). These are the eight components, or subtests, that appear in the sample EGRA exam in English.

| Component | Early reading skill | Skill demonstrated by students' ability to |
|---|---|---|
| 1. Letter name knowledge | Letter recognition | Provide the name of upper- and lowercase letters in random order. |
| 2. Phonemic awareness | Phonemic awareness | Segment words into phonemes.<br>Identify the initial sounds in different words. |
| 3. Letter sound knowledge | Phonics | Provide the sound of upper- and lowercase letters distributed in random order. |
| 4. Familiar word reading | Word reading | Read simple and common one- and two-syllable words. |
| 5. Unfamiliar nonword reading | Alphabetic principle | Make grapheme-phoneme correspondences (GPCs) through the reading of simple nonsense words. |
| 6a. Oral reading fluency with comprehension | Oral reading fluency | Read a text with accuracy, with little effort, and at a sufficient rate. |
| 6b. Oral reading fluency with comprehension | Reading Comprehension | Respond correctly to different types of questions including literal and inferential questions about the text they have read. |
| 7. Listening comprehension | Listening comprehension | Respond correctly to different types of questions including literal and inferential questions about the text the enumerator reads to them. |
| 8. Dictation | Alphabetic principle | Write, spell, and use grammar properly through a dictation exercise. |

# Appendix B

## Sample Evaluator Sheet

Appears in *EGRA Toolkit* (RTI International, 2009, p. 24).

### *Sample Assessment Design: Letter Name Knowledge*

*Show the child the sheet of letters in the student stimuli booklet. Say:*

**Here is a page full of letters of the alphabet. Please tell me the NAMES of as many letters as you can--not the SOUNDS of the letters, but the names.**

**For example, the name of this letter** *[point to A]* **is "A"**

**Let's practise: tell me the name of this letter** *[point to V]*:

*If the child responds correctly say:* **Good, the name of this letter is "VEE."**
*If the child does not respond correctly, say:* **The name of this letter is "VEE."**

**Now try another one: tell me the name of this letter** *[point to L]*:

*If the child responds correctly say:* **Good, the name of this letter is "ELL."**
*If the child does not respond correctly, say:* **The name of this letter is "ELL."**

**Do you understand what you are to do?**

**When I say "Begin," please name the letters as quickly and carefully as you can. Start here and continue this way.** *[Point to the first letter on the row after the example and draw your finger across the first line].* **If you come to a letter you do not know, I will tell it to you. Otherwise I will keep quiet & listen to you. Ready? Begin.**

*Start the timer when the child reads the first letter. Follow along with your pencil and* ***clearly*** *mark any incorrect letters with a slash ( / ). Count self-corrections as correct. If you've already marked the self-corrected letter as incorrect, circle the letter and go on.* ***Stay quiet,*** *except when providing answers as follows: if the child hesitates for 3 seconds, provide the name of the letter, point to the next letter and say "***Please go on***." Mark the letter you provide to the child as incorrect. If the student gives you the letter sound, rather than the name, provide the letter name and say: ["***Please tell me the NAME of the letter***"]. This prompt may be given only once during the exercise.*

***AFTER 60 SECONDS SAY, "stop." Mark the final letter read with a bracket ( ] ).***
***Early stop rule:*** *If the child does not give a single correct response on the first line, say "***Thank you!***", discontinue this exercise, check the box at the bottom, and go on to the next exercise.*

*Example :* A v L

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
|---|---|---|---|---|---|---|---|---|---|---|
| L | i | h | R | S | y | E | O | n | T | (10) |
| i | e | T | D | A | t | a | d | e | w | (20) |
| h | O | e | m | U | r | L | G | R | u | (30) |
| g | R | B | E | i | f | m | t | s | r | (40) |
| S | T | C | N | p | A | F | c | a | E | (50) |
| y | s | Q | A | M | C | O | t | n | P | (60) |
| e | A | e | s | O | F | h | u | A | t | (70) |
| R | q | H | b | S | i | g | m | i | L | (80) |
| L | i | N | O | e | o | E | r | p | X | (90) |
| N | A | c | D | d | I | O | j | e | n | (100) |

*Time remaining on stopwatch at completion (number of SECONDS) :* ☐

*Check this box if the exercise was discontinued because the child had no correct answers in the first line.* ☐

# Appendix C

## Installing and Configuring Packaged Software Used in ICT4EGRA

The ICT4EGRA system relies heavily on preexisting packaged software. Further instructions on installing and configuring this software can be found at the links below:

| | |
|---|---|
| Open Data Kit Aggregate | http://opendatakit.org/use/aggregate/ |
| Open Data Kit Collect | http://opendatakit.org/use/collect/ |
| Google App Engine | http://code.google.com/appengine/ |
| Google Android SDK | http://developer.android.com/sdk/index.html |
| Google Spreadsheets | http://docs.google.com/spreadsheets |
| Google Apps Script | http://code.google.com/googleapps/appsscript/articles/yourfirstscript.html |

# Appendix D

## `C1_LetterNameKnowledge.xml` (Sample XForms Component Source Code)

An XForms .xml file for component one (Letter Name Knowledge) of the sample EGRA instrument provided by RTI International in the *EGRA Toolkit* (2009).

```
<?xml version="1.0"?>
<h:html xmlns="http://www.w3.org/2002/xforms" xmlns:h="http://www.w3.org/1999/xhtml"
xmlns:ev="http://www.w3.org/2001/xml-events"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:jr="http://openrosa.org/javarosa">
<h:head>
 <h:title>Component 1: Letter Name Knowledge</h:title>
  <model>
   <instance>
    <data id="C1_LetterNameKnowledge_v1">
     <C1_Instructions/>
     <C1_Assessment/>
     <C1_LastLetter/>
     <C1_Time>0</C1_Time>
    </data>
   </instance>
   <itext>
    <translation lang="eng">
     <text id="/data/C1_Instructions:label">
      <value>Component 1: Letter Name Knowledge</value>
     </text>
    <text id="/data/C1_Instructions:hint">
     <value>Start the timer when the child reads the first letter. Follow along with
the child and check correct words.</value>
    </text>
    <text id="/data/C1_Assessment:label">
     <value>Select all correctly pronounced letters.</value>
    </text>
    <text id="/data/C1_Assessment:hint">
     <value>Remember the number of the last character read.</value>
    </text>
    <text id="/data/C1_LastLetter:label">
     <value>Enter the number of the last character spoken</value>
    </text>
    <text id="/data/C1_LastLetter:hint">
     <value>Enter "0" for early finish; "100" if the student read all letters</value>
    </text>
    <text id="/data/C1_Time:label">
      <value>Enter the time left on the stopwatch in seconds</value>
    </text>
    <text id="/data/C1_Time:hint">
      <value>Range: 0 to 60 seconds (enter 0 seconds if the student did not
      finish)</value>
    </text>
    <text id="/data/C1_Assessment:letter01">
     <value>01: L</value>
    </text>
    <text id="/data/C1_Assessment:letter02">
     <value>02: i</value>
    </text>
        ...omitted...
    <text id="/data/C1_Assessment:letter100">
     <value>100: n</value>
```

```
    </text>
   </translation>
   </itext>
   <bind nodeset="/data/C1_Instructions" type="int" readonly="true()"/>
   <bind nodeset="/data/C1_Assessment" type="select"/>
   <bind nodeset="/data/C1_LastLetter" type="int" required="true()" constraint="(.
&gt;= 0 and . &lt;= 100)"/>
   <bind nodeset="/data/C1_Time" type="int" constraint="(. &gt;= 0 and . &lt;= 60)"
required="true()"/>
  </model>
 </h:head>
 <h:body>
  <input ref="/data/C1_Instructions">
   <label ref="jr:itext('/data/C1_Instructions:label')"/>
   <hint ref="jr:itext('/data/C1_Instructions:hint')"/>
  </input>
  <select ref="/data/C1_Assessment">
   <label ref="jr:itext('/data/C1_Assessment:label')"/>
   <hint ref="jr:itext('/data/C1_Assessment:hint')"/>
   <item>
    <label ref="jr:itext('/data/C1_Assessment:letter01')"/>
    <value>01</value>
   </item>
       ...omitted...
   <item>
    <label ref="jr:itext('/data/C1_Assessment:letter100')"/>
    <value>100</value>
   </item>
  </select>
  <input ref="/data/C1_LastLetter">
   <label ref="jr:itext('/data/C1_LastLetter:label')"/>
   <hint ref="jr:itext('/data/C1_LastLetter:hint')"/>
  </input>
  <input ref="/data/C1_Time">
   <label ref="jr:itext('/data/C1_Time:label')"/>
   <hint ref="jr:itext('/data/C1_Time:hint')"/>
  </input>
 </h:body>
</h:html>
```

# Appendix E

## C1_UpdateLetters.sh

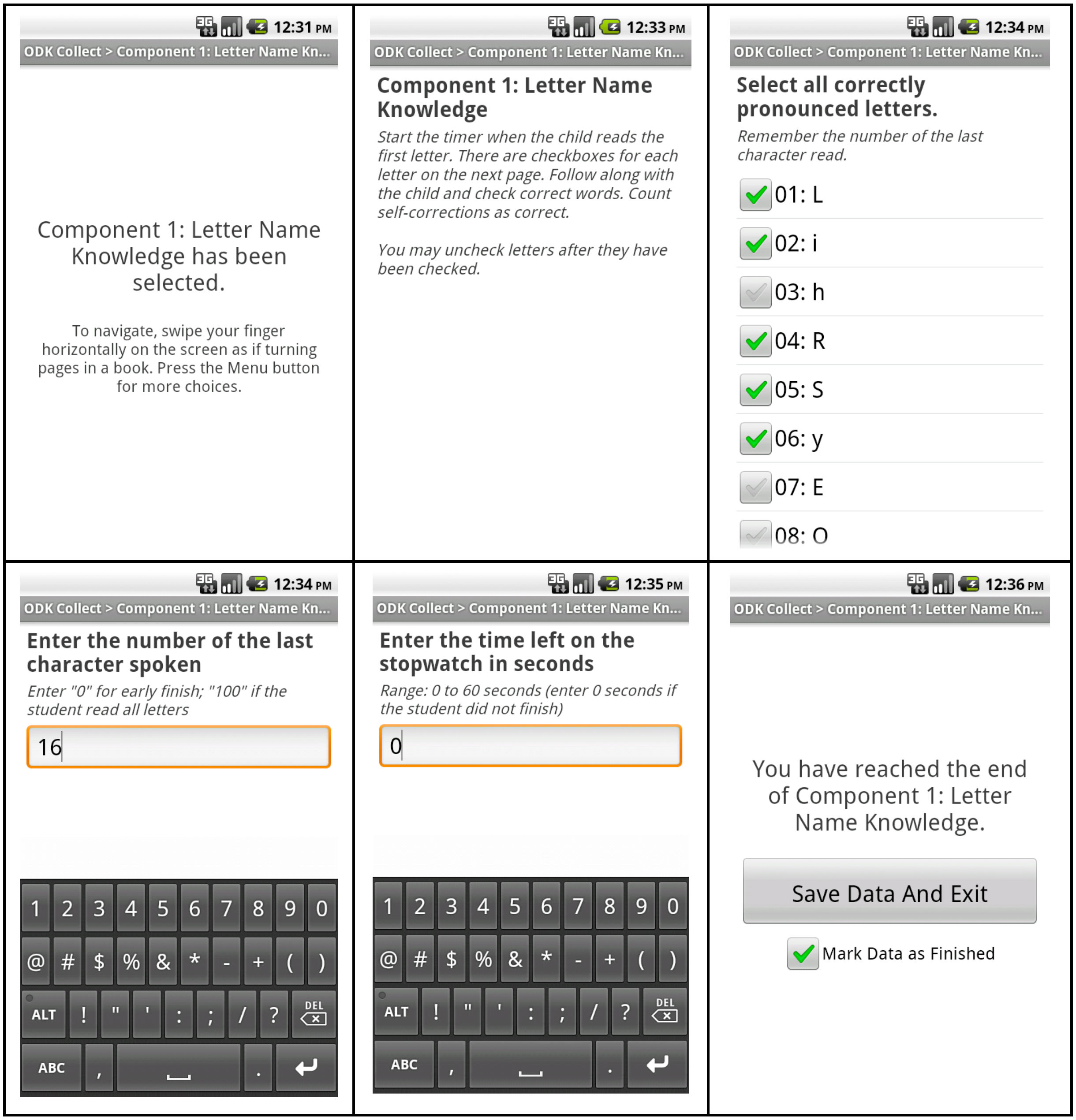
A shell script for updating letters in the component one (Letter Name Knowledge) XForms file.

```
#!/bin/sh

# USAGE
# ./C1_UpdateLetters.sh file1 file2

# USAGE EXAMPLE
# ./C1_UpdateLetters.sh NewLetters.txt C1_LetterNameKnowledge.xml

# DESCRIPTION
# Replaces the letters in a Letter Name Knowledge form.

# ARGUMENTS
# $1 = file1 = a text file with 100 letters separated by spaces
# $2 = file2 = an EGRA Letter Knowledge XForms survey

for (( i=1; i<=100; i++ ))
do
       l=$(awk '{print $x}' x=$i $1) # read the ith letter from the input file
       if [ $i -lt 10 ]
       then
    echo "0$i = $l"
    sed -i temp 's/>0'"$i"': .</>0'"$i"': '"$l"'</' $2
       else
    echo "$i = $l"
    sed -i temp 's/>'"$i"': .</>'"$i"': '"$l"'</' $2
       fi
done
```

# Appendix F

## Screenshots of Letter Name Knowledge XForms Survey

Screenshots of the complete XForms survey for component one as rendered by ODK Collect.



# Appendix G

## addComponents.sh

A shell script which automates the process of combining XForms files for each component into a single, larger XForms file.

```
#!/bin/sh

# USAGE
# ./addComponents.sh file1 file2 ...

# USAGE EXAMPLE
# ./addComponents.sh EGRA_template.xml C0_StudentInformation.xml
C1_LetterNameKnowledge.xml

# DESCRIPTION
# Adds the components described in file2, file3, to fileN to the EGRA assessment in
file1.

egra_file="$1"
shift # now the remaining arguments are all component forms to be added

# finds the line number of the instance of the argument in the component file
# args: (regex to find)
function get_line_number () {
	echo $(egrep -n "$1" "$component_file" | awk '{print $1}' | sed 's/://')
}

# add appends the argument as a line to the current section in the egra_file
# reqs: $section must be set
# args: (line to add)
function add_line ()
{
	# the -i flag adds whitespace to the front of the line
	if [ "$1" = "-i" ]; then
	 line=$2
	 ws="\1" # backreference whitespace in sed command
	else # don't add whitespace
	 line="$1"
	 ws=""
	fi
	echo "Adding:" "$line"
	line=$(echo "$line" | sed 's/[\/\\\&]/\\&/g') # escape chars
	sed -Ei tmp 's/^([ \'$'\t'']*)<!--'"$section"'-->/'$ws''"$line"'\'$'\n''&/'
"$egra_file"
}

# copies a section from the component file to the EGRA file
# requires $section, $start, and $end to be set properly
function add_section ()
{
	add_line -i "<!--START:"$id"-->"
	# add group tag and label for body section
	if [ "$section" = "body" ]; then
	 add_line -i "<group>"
	 add_line -i "<label>$label</label>"
	fi
```

```
	for (( i=$start+1; i<$end; i++ ))
	do
	 line=$(sed -n $i,"$i"p "$component_file") # read a line
	 if [ "$section" = "body" ]; then
	  add_line "   $line" # now nested in the <group> tag
	 else
	  add_line "$line"
	 fi
	done
	# add closing group tag for body section
	if [ "$section" = "body" ]; then
	 add_line -i "</group>"
	fi
	add_line -i "<!--END:"$id"-->"
}

function add_component ()
{
	# extract id and label from the component.xml file
	id=$(sed -En 's/^.*<data id="([^"]*)">/\1/p' "$component_file" | tr -d "\r")
	label=$(sed -En 's/^.*<h:title>([^<]*)<\/h:title>/\1/p' "$component_file" | tr
-d "\r")

	# copy the data section
	section=data
	start=$(get_line_number "<data")
	end=$(get_line_number "</data>")
	add_section

	# copy the translation section
	section=translation
	start=$(get_line_number "<translation")
	end=$(get_line_number "</translation")
	add_section

	# copy the bind section
	section=bind
	start=$(get_line_number "</itext")
	end=$(get_line_number "</model")
	add_section

	# copy the group section
	section=body
	start=$(get_line_number "<h:body>")
	end=$(get_line_number "</h:body>")
	add_section
}

while (( "$#" ))
do
	component_file="$1"
	add_component
	shift
done

# remove carriage returns
tr -d "\r" < "$egra_file" > "$egra_file"tmp
mv "$egra_file"tmp "$egra_file"
```

# Appendix H

## EGRA_template.xml

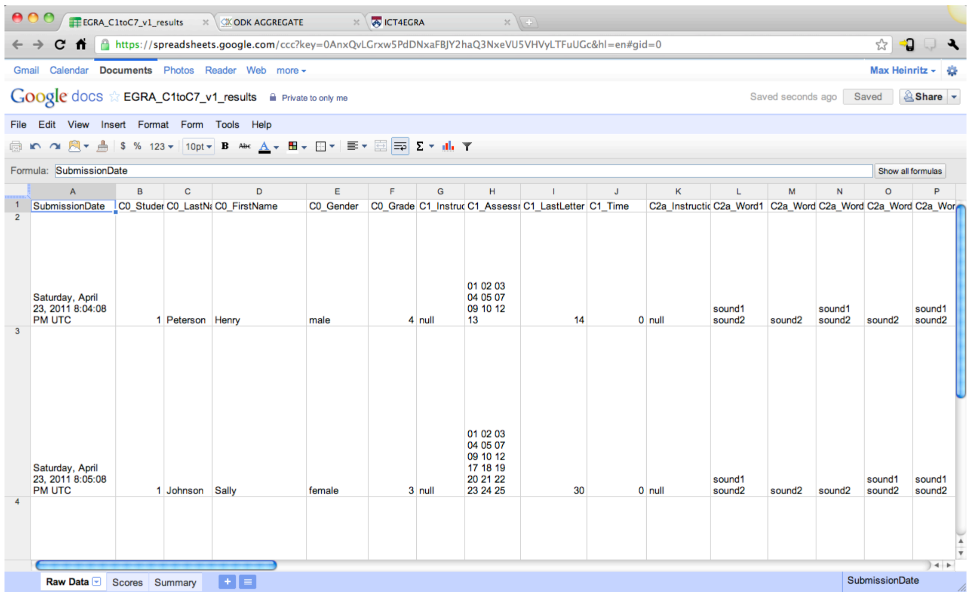
This is a template XForms file for an EGRA instrument. The addComponents.sh script in Appendix F can be used to add component XForms source code to this file to create complete EGRA XForms survey.

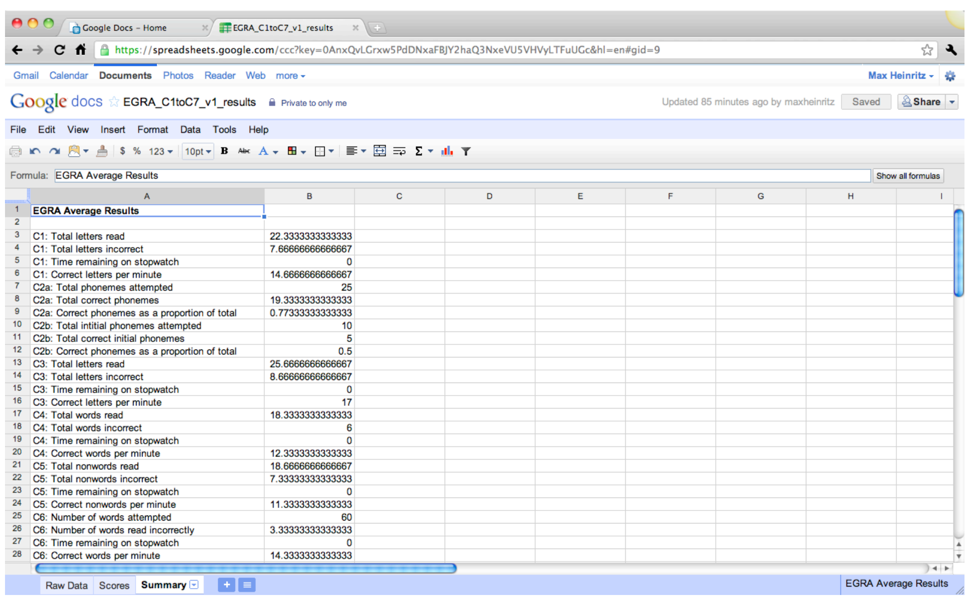
```
<?xml version="1.0"?>
<h:html xmlns="http://www.w3.org/2002/xforms" xmlns:h="http://www.w3.org/1999/xhtml"
xmlns:ev="http://www.w3.org/2001/xml-events"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:jr="http://openrosa.org/javarosa">
  <h:head>
       <h:title>EGRA v1</h:title>
       <model>
        <instance>
         <data id="EGRA_v1">
          <!--data-->
         </data>
        </instance>
        <itext>
         <translation lang="eng">
          <!--translation-->
         </translation>
        </itext>
        <!--bind-->
       </model>
  </h:head>
  <h:body>
   <!--body-->
  </h:body>
</h:html>
```

# Appendix I

## Screenshot of Google Spreadsheet

This is the Google Spreadsheet used analyze the EGRA results. Raw data can be imported using the method described by Google (2011).



# Appendix J

## analyzeEGRAResults()

A Google Apps Script file for analysis of the raw EGRA data. The script parse and scores the raw data for each component.

```
function findCol(sheet,fieldName) {
  var data = sheet.getDataRange().getValues();
  // find the column with the title "fieldName"
  for( var i in data[0] )
       if( fieldName == data[0][i] )
       // get the range in that column from row 2 until the bottom
       return parseInt(i)+1;
  return null;
}

function addStudentInformation(raw, scores, row, col) {
  if (row==1) {
       scores.getRange(row,col++).setValue("Student ID");
       scores.getRange(row,col++).setValue("Last Name");
       scores.getRange(row,col++).setValue("First Name");
       scores.getRange(row,col++).setValue("Gender");
       scores.getRange(row,col++).setValue("Grade");
       return col;
  }
  var dataCol = findCol(raw, "C0_StudentID");
  var id = raw.getRange(row, dataCol++).getValue();
  var last = raw.getRange(row, dataCol++).getValue();
  var first = raw.getRange(row, dataCol++).getValue();
  var gender = raw.getRange(row, dataCol++).getValue();
  var grade = raw.getRange(row, dataCol++).getValue();
  scores.getRange(row,col++).setValue(id);
  scores.getRange(row,col++).setValue(last);
  scores.getRange(row,col++).setValue(first);
  scores.getRange(row,col++).setValue(gender);
  scores.getRange(row,col++).setValue(grade);
  return col;
}

function scoreTimedComponent(componentNum, raw, scores, row, col) {
  if (row==1) {
       var items = "letters";
       if (componentNum == 4)
       items = "words";
       if (componentNum == 5)
       items = "nonwords";
       scores.getRange(row,col++).setValue("C"+componentNum+": Total "+items+" read");
       scores.getRange(row,col++).setValue("C"+componentNum+": Total "+items+"
incorrect");
       scores.getRange(row,col++).setValue("C"+componentNum+": Time remaining on
stopwatch");
       scores.getRange(row,col++).setValue("C"+componentNum+": Correct "+items+" per
minute");
       return col;
  }
  var dataCol = findCol(raw, "C"+componentNum+"_Assessment");
  // find the number of items checked
  numOfItems = 0; // items could be letters, words, or nonwords
```

```
  var spaced = raw.getRange(row, dataCol++).getValue();
  if (spaced) // might be empty if zero items were checked
       numOfItems = spaced.split(" ").length;
  lastItem = raw.getRange(row, dataCol++).getValue();
  seconds = raw.getRange(row, dataCol++).getValue();
  if (seconds=="null" || seconds.length==0)
       seconds = 0;

  // if checked values are correct
  scores.getRange(row,col++).setValue(lastItem);
  scores.getRange(row,col++).setValue(lastItem-numOfItems);
  scores.getRange(row,col++).setValue(seconds);
  scores.getRange(row,col++).setValue((numOfItems)/((60 - seconds)/60));

  return col;
  /* if checked letters are incorrect
  scores.getRange(row,col++).setValue(lastItem);
  scores.getRange(row,col++).setValue(numOfItems);
  scores.getRange(row,col++).setValue(seconds);
  scores.getRange(row,col++).setValue((lastItem-numOfItems)/((60 - seconds)/60)); */
}

function scoreC2a(raw, scores, row, col) {
  // 2*5 + 3*5 = 25 phonemes total (5 2-phoneme words, 5 3-phoneme words)
  var NUM_TOTAL_PHONEMES = 25;

  if (row==1) {
       scores.getRange(row,col++).setValue("C2a: Total phonemes attempted");
       scores.getRange(row,col++).setValue("C2a: Total correct phonemes");
       scores.getRange(row,col++).setValue("C2a: Correct phonemes as a proportion of
total");
       return col;
  }
  var dataCol = findCol(raw, "C2a_Word1");
  // the next ten columns are all words
  var phonemes = raw.getRange(row, dataCol, 1, 10).getValues();
  var totalChecked = 0;
  for ( var i=0; i<10; i++ ) {
       if(phonemes[0][i]) // may be null if empty
       totalChecked+=phonemes[0][i].split(" ").length;
  }
  // if checked letters are correct
  scores.getRange(row,col++).setValue(NUM_TOTAL_PHONEMES);
  scores.getRange(row,col++).setValue(totalChecked);
  scores.getRange(row,col++).setValue(totalChecked/NUM_TOTAL_PHONEMES);
  return col;

  /* if checked letters are incorrect
  scores.getRange(row,col++).setValue(NUM_TOTAL_PHONEMES);
  scores.getRange(row,col++).setValue(NUM_TOTAL_PHONEMES-totalChecked);
  scores.getRange(row,col++).setValue((NUM_TOTAL_PHONEMES-
totalChecked)/NUM_TOTAL_PHONEMES);
  return col; */
}

function scoreC2b(raw, scores, row, col) {
  var NUM_TOTAL_WORDS = 10;
  if (row==1) {
       scores.getRange(row,col++).setValue("C2b: Total intitial phonemes attempted");
       scores.getRange(row,col++).setValue("C2b: Total correct initial phonemes");
       scores.getRange(row,col++).setValue("C2b: Correct phonemes as a proportion of
total");
       return col;
```

```
  }
  var dataCol = findCol(raw, "C2b_Word1");
  var phonemes = raw.getRange(row, dataCol, 1, 10).getValues();
  var totalCorrect = 0;
  for ( var i=0; i<10; i++ ) {
       if(phonemes[0][i]=="correct")
       totalCorrect++;
  }
  scores.getRange(row,col++).setValue(NUM_TOTAL_WORDS);
  scores.getRange(row,col++).setValue(totalCorrect);
  scores.getRange(row,col++).setValue(totalCorrect/NUM_TOTAL_WORDS);
  return col;
}

function scoreC6(raw, scores, row, col) {
  var NUM_WORDS_IN_STORY = 60;
  var NUM_QUESTIONS = 4;

  if (row==1) {
       scores.getRange(row,col++).setValue("C6: Number of words attempted");
       scores.getRange(row,col++).setValue("C6: Number of words read incorrectly");
       scores.getRange(row,col++).setValue("C6: Time remaining on stopwatch");
       scores.getRange(row,col++).setValue("C6: Correct words per minute");
       scores.getRange(row,col++).setValue("C6: Comprehesion questions asked");
       scores.getRange(row,col++).setValue("C6: Comprehesion questions answered
correctly");
       return col;
  }
  var dataCol = findCol(raw, "C6_PassageWords");
  var spaced = raw.getRange(row, dataCol++).getValue();
  var numOfCheckedWords = 0;
  if (spaced) // might be empty if zero words were checked
       numOfCheckedWords = spaced.split(" ").length;
  var lastWord = raw.getRange(row, dataCol++).getValue();
  var secondsRemaining = raw.getRange(row, dataCol++).getValue();
  var answers = raw.getRange(row, dataCol, 1, NUM_QUESTIONS).getValues();
  var correct = 0;
  var attempted = 0;
  // four possible responses for each question: correct, incorrect, no response, or
not read
  for ( var i=0; i<NUM_QUESTIONS; i++ ) {
       if (answers[0][i]=="correct") {
       correct++;
       attempted++;
       }
       else if (answers[0][i]=="incorrect")
       attempted++;
       else if (answers[0][i]=="noResponse")
       attempted++;
  }

  scores.getRange(row,col++).setValue(NUM_WORDS_IN_STORY);
  scores.getRange(row,col++).setValue(lastWord-numOfCheckedWords);
  scores.getRange(row,col++).setValue(secondsRemaining);
  scores.getRange(row,col++).setValue(numOfCheckedWords/((60-secondsRemaining)/60));
  scores.getRange(row,col++).setValue(attempted);
  scores.getRange(row,col++).setValue(correct);
  return col;
}

function scoreC7(raw, scores, row, col) {
  var NUM_QUESTIONS = 4;
```

```
  if (row==1) {
       scores.getRange(row,col++).setValue("C7: Number of questions asked");
       scores.getRange(row,col++).setValue("C7: Number of questions answered
correctly");
       return col;
  }
  var dataCol = findCol(raw, "C7_Question1");
  var answers = raw.getRange(row, dataCol, 1, NUM_QUESTIONS).getValues();
  var correct = 0;
  for ( var i=0; i<NUM_QUESTIONS; i++ ) {
       if (answers[0][i]=="correct")
       correct++;
  }

  scores.getRange(row,col++).setValue(NUM_QUESTIONS);
  scores.getRange(row,col++).setValue(correct);
  return col;
}

function buildScoresSheet(raw,scores) {
  scores.clear();

  // find the number of entries we need to loop through
  var data = raw.getDataRange().getValues();
  var numStudents = data.length;
  var data = raw.getDataRange().getValues();
  for( var i in data )
       if( data[i][0].length == 0 ) {
       lastSubmissionRow = i;
       break;
       }

  for ( var row=1; row<=numStudents; row++ ) {
       col = 1;
       col=addStudentInformation(raw, scores, row, col); // Student information
       col=scoreTimedComponent(1,raw, scores, row, col); // C1
       col=scoreC2a(raw, scores, row, col); // C2a
       col=scoreC2b(raw, scores, row, col); // C2b
       col=scoreTimedComponent(3,raw, scores, row, col); // C3
       col=scoreTimedComponent(4,raw, scores, row, col); // C4
       col=scoreTimedComponent(5,raw, scores, row, col); // C5
       col=scoreC6(raw, scores, row, col); // C6
       col=scoreC7(raw, scores, row, col); // C7
  }

  // how many columns until the first data column?
  return addStudentInformation(raw, scores, 1, 1);
}

function buildSummarySheet(scores,summary,firstColumnWithData) {
  summary.clear();

  summary.getRange(1,1).setValue("EGRA Average Results");
  summary.getRange(1,1).setFontWeight("bold");

  var rows = scores.getDataRange().getValues().length;
  var fields = scores.getDataRange().getValues()[0].length;
  // find the column with the title "fieldName"
  for( col = firstColumnWithData; col<=fields; col++ ) {
       var curRow = col-firstColumnWithData+3;
       var title = scores.getRange(1,col).getValue();
       var range = scores.getRange(2,col,rows);
       scores.getParent().setNamedRange("scoresColumn"+col,range);
```

```
        summary.getRange(curRow,1).setValue(title);
        summary.getRange(curRow,2).setValue("=AVERAGE(scoresColumn"+col+")");
  }
}

function analyze() {
  var spreadsheet = SpreadsheetApp.getActiveSpreadsheet();
  var rawDataSheet = spreadsheet.getSheets()[0];

  var scoresSheet = spreadsheet.getSheetByName("Scores");
  if (!scoresSheet)
       scoresSheet = spreadsheet.insertSheet("Scores");
  var firstColWithData = buildScoresSheet(rawDataSheet,scoresSheet);

  var summarySheet = spreadsheet.getSheetByName("Summary");
  if (!summarySheet)
       summarySheet = spreadsheet.insertSheet("Summary");
  buildSummarySheet(scoresSheet,summarySheet,firstColWithData);
}
```