Open Data Kit: Building Mobile Application Frameworks for Disconnected Data Management

Waylon Brunette
University of Washington

Open Data Kit has a team of contributors and my research would not be possible without the team’s hard work.
Organizations working in developing regions rely on field workers to collect data.
Paper Methods

- Paper-based forms – ad hoc design
- Long time-lag to usable data
- Little or no historical data
Paper Systems are difficult to Search and Transport
- Hard to search
  - “Find the patient record for ‘Hartung’ ”

- Impossible to query
  - “How many patients presented with Malaria last month? ”

- Hard to share
  - What if patient goes to a different clinic?
e-IMCI on a PDA

Brian DeRenzi, Neal Lesh, Tapan Parikh, Clayton Sims, Werner Maokla, Mwajuma Chemba, Yuna Hamisi, David S hellenberg, Marc Mitchell, and Gaetano Borriello.

Mobile Device Expansion

Mobile Device availability **DOES NOT** equate to Internet Access

Limited Internet Access

a. ICT access by population

- Total global population: ~7.4 billion
- Within mobile coverage: 7 billion
- Mobile phones: 5.2 billion
- Total internet users: 3.2 billion
- High-speed internet: 1.1 billion

b. A closer look at the world’s offline population

- Countries outside of the top 20
- Total internet users: 3.2 billion
- High-speed internet: 1.1 billion
- Details on specific populations in various countries listed.


Note: High-speed internet (broadband) includes the total number of fixed-line broadband subscriptions (such as DSL, cable modems, fiber optics), and the total number of 4G/LTE mobile subscriptions, minus a correcting factor to allow for those who have both types of access. 4G = fourth generation; DSL = digital subscriber line; ICT = information and communication technology; LTE = Long Term Evolution.

Limited Internet Access

**UPDATE: The India Government in 2019 estimates the offline users to be over 700 million.**


Note: High-speed internet (broadband) includes the total number of fixed-line broadband subscriptions (such as DSL, cable modems, fiber optics), and the total number of 4G/LTE mobile subscriptions, minus a correcting factor to allow for those who have both types of access. 4G = fourth generation; DSL = digital subscriber line; ICT = information and communication technology; LTE = Long Term Evolution.


**Update from: The Indian Telecom Services Performance Indicators. https://main.trai.gov.in/sites/default/files/PIR08012019.pdf, January 2019**
Paper Systems are difficult to Search and Transport

- Hard to search
  - “Find the patient record for ‘Hartung’ ”

- Impossible to query
  - “How many patients presented with Malaria last month? ”

- Hard to share
  - What if patient goes to a different clinic?
Platform Shift from PCs to

Smart Phone + Cloud
1. Build form

2. Collect data

3. Aggregate results

http://opendatakit.org

GOAL: Magnify human resources through technology

Open Data Kit (ODK)

• First release in 2009 (started in 2008)
• Mobile data collection tools for Android devices
• Modular, open architecture
• Open source (Apache 2 license)

• KEY FEATURES TO SUCCESS:
  – Domain Independent Tools
  – Disconnected Operation

GOAL: Magnify human resources through technology
ODK Videos

• Surui
  – http://www.youtube.com/watch?v=_gKkYc9ntHQ

• Reproductive Health Vouchers
  – http://vimeo.com/38123850
Over 85% of surveyed ODK users reported rural deployments

(Cobb & Sudar**)

Open Data Kit

• Make tools highly modular and customizable
  – Enables organizations to compose tools that are appropriate for their deployments

• Exploit open interfaces and standards
  – Avoid “silo-ed” monolithic proprietary solutions

• Allow organizations to leverage evolving technologies
  – Avoid early obsolescence
Health Model

Cloud infrastructure and data storage

Tasks

Data

CHW Supervisors

Health Clinic

Ministry of Health
Generic Model

Cloud infrastructure and data storage

Forest Wardens
Response Teams
Environment Ministry

data

tasks
ODK Collect

Automated Survey Renderer with enhanced data types
- Pictures, Video
- GPS
- Barcode
XForms

Describes the Form logic and Data Schema

<?xml version="1.0"?>
  <h:head>
    <h:title>Geo Tagger</h:title>
  </h:head>
  <model>
    <instance>
      <geotagger id="geo_tagger1">
        <timestamp />
        <device_id/>
        <subscriber_id/>
        <image/>
        <geopoint/>
        <string/>
      </geotagger>
    </instance>
    <bind nodeset="/geotagger/timestamp" type="dateTime" jr:preload="timestamp" jr:preloadParams="start"/>
    <bind nodeset="/geotagger/device_id" type="string" jr:preload="property" jr:preloadParams="deviceid"/>
    <bind nodeset="/geotagger/subscriber_id" type="string" jr:preload="property" jr:preloadParams="subscriberid"/>
    <bind nodeset="/geotagger/geopoint" type="geopoint"/>
    <bind nodeset="/geotagger/image" type="binary"/>
    <bind nodeset="/geotagger/string" type="string"/>
  </model>
  <h:body>
    <upload ref="image" mediatype="image/*">
      <label>What do you see?</label>
    </upload>
    <input ref="geopoint">
      <label>Where are you?</label>
    </input>
    <input ref="string">
      <label>Any other thoughts?</label>
    </input>
  </h:body>
</h:html>
Building XForms

- **ODK Build**
  - Drop-n-Drag UI for creating form

- **ODK XLSForm**
  - Use a Excel to enter form information
  - Tool transforms the xls file into Xform
ODK Aggregate

Stores or forwards data to external systems

---

<table>
<thead>
<tr>
<th>Image</th>
<th>Location Latitude</th>
<th>Location Longitude</th>
<th>Location Altitude</th>
<th>Location Accuracy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.jpg" alt="Image" /></td>
<td>47.65424976-122.3046976</td>
<td>21.29999924</td>
<td>6.708204</td>
<td>HUB construction</td>
<td></td>
</tr>
<tr>
<td><img src="image2.jpg" alt="Image" /></td>
<td>47.6484739-122.29989659</td>
<td>20.39999924</td>
<td>6.568654</td>
<td>Docks at WAC</td>
<td></td>
</tr>
<tr>
<td><img src="image3.jpg" alt="Image" /></td>
<td>47.65365942-122.3256423</td>
<td>-14.89000062</td>
<td>6.708204</td>
<td>foot of Latona</td>
<td></td>
</tr>
<tr>
<td><img src="image4.jpg" alt="Image" /></td>
<td>47.64854424-122.33644095</td>
<td>-7.09000099</td>
<td>6.324064</td>
<td>home</td>
<td></td>
</tr>
<tr>
<td><img src="image5.jpg" alt="Image" /></td>
<td>47.64540379-122.33636568</td>
<td>-21.79999961</td>
<td>4.472136</td>
<td>Kite hill... Gasworks</td>
<td></td>
</tr>
<tr>
<td><img src="image6.jpg" alt="Image" /></td>
<td>47.62708792-122.33274967</td>
<td>-8.0</td>
<td>6.708204</td>
<td>Chandler's Cove</td>
<td></td>
</tr>
<tr>
<td><img src="image7.jpg" alt="Image" /></td>
<td>47.65382488-122.31314593</td>
<td>31.0</td>
<td>5.0</td>
<td>The Ave</td>
<td></td>
</tr>
<tr>
<td><img src="image8.jpg" alt="Image" /></td>
<td>47.69942725-122.32058416</td>
<td>25.39999962</td>
<td>5.0</td>
<td>Ravenna Park</td>
<td></td>
</tr>
<tr>
<td><img src="image9.jpg" alt="Image" /></td>
<td>47.68088888-122.3291259</td>
<td>48.69999847</td>
<td>5.0</td>
<td>Greensite</td>
<td></td>
</tr>
<tr>
<td><img src="image10.jpg" alt="Image" /></td>
<td>47.68999484-122.35543011</td>
<td>88.69999847</td>
<td>7.211925</td>
<td>Greenwood</td>
<td></td>
</tr>
<tr>
<td><img src="image11.jpg" alt="Image" /></td>
<td>47.60081164-122.36037847</td>
<td>80.0</td>
<td>5.0</td>
<td>Panhandle Ridge</td>
<td></td>
</tr>
<tr>
<td><img src="image12.jpg" alt="Image" /></td>
<td>47.60111747-122.38012042</td>
<td>66.80000305</td>
<td>5.0</td>
<td>Upper Fremont</td>
<td></td>
</tr>
<tr>
<td><img src="image13.jpg" alt="Image" /></td>
<td>47.64886686-122.36156679-13.80000319</td>
<td>6.708204</td>
<td>Freemont Bridge</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Effort (yellow) and threats (red) collected by village forest monitors using ODK around Gombe National Park.
Deployment Concerns

Network connectivity is a *PERSISTENT* concern

**ORGANIZATION’S CONCERNS:**

- Need to be able to share data between devices
- Often sparse connectivity in rural environments
- Type of connectivity varies by location
- Data transmission costs (can be high)
- Administrative concerns restrict how data can be transmitted or stored

**Limiting factors**

- Power
- Cost
- Expertise

Expanding & Refining ODK

- ODK 1 deployed successfully around the world
- Lots of requests for more features & expansion

ODK 1

(DUCES: less challenging, forms, uni-directional, disconnected)

versus

ODK X (formerly ODK 2)

(DUCES: more challenging, bi-directional, more expertise) - FLEXIBLE

CASE STUDY: EU Refugee Crisis

Photo Courtesy of the International Federation of Red Cross and Red Crescent Societies
CASE STUDY: EU Refugee Crisis

Photo Courtesy of the International Federation of Red Cross and Red Crescent Societies
**CASE STUDY: World Mosquito Program**

- **World Mosquito Program** - uses naturally occurring bacteria (*Wolbachia*) to reduce the ability of mosquitoes to transmit viruses (e.g., dengue, chikungunya, Zika)

- **Using ODK-X** in Brazil, Columbia, Indonesia, Australia, and Vietnam

- Program Manager Feedback:
  - “*quite easy to use and we haven’t had any acceptance issues.*”
  - “*the app is scaling quite well*”
CASE STUDY: HIV Patient Tracking

• Adaptive Strategies for Preventing and Treating Lapses of Retention in Care (AdaPT-R)
  – UCSF Randomized Control Trial in Kenya

• ODK-X deployed in 5 clinics for multiple years
  – Clinics serve 65,000 patients
  – 17,000 HIV Patients
  – 18 clinical employees using ODK 2.0

"We needed a solution for capturing data from multiple forms and that would allow longitudinal follow-up of individual patients. We had experience with earlier versions of ODK, so the new features of 2.0 made it the only option for us if we wanted phone-based longitudinal form completion. Would definitely recommend ODK 2.0!“  - Primary Investigator
ODK-X Case Studies

• ODK-X had an iterative requirements gathering process
  – Surveys
  – Pilot deployments in 18+ countries by a variety of organizations,
  – The ODK-X tool suite went through a significant redesign from the original ODK-X vision

• To validate the derived requirements we examined 6 case studies

<table>
<thead>
<tr>
<th>Table 3: Case Study ODK 2.0 Feature Requirement Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex / Non-Linear Workflows</td>
</tr>
<tr>
<td>Link Longitudinal Data To Collected Data</td>
</tr>
<tr>
<td>Data Security and User Permissions</td>
</tr>
<tr>
<td>Reuse of Data Fields Across Forms</td>
</tr>
<tr>
<td>Bidirectional Synchronization</td>
</tr>
<tr>
<td>Customizable Form Presentation</td>
</tr>
<tr>
<td>Custom JavaScript Apps</td>
</tr>
<tr>
<td>Sensor Integration</td>
</tr>
<tr>
<td>Paper Digitization</td>
</tr>
<tr>
<td>Custom Data Types Update Multiple Fields in a Single User Action</td>
</tr>
</tbody>
</table>
Missing capabilities

• Updating data on the mobile device
  – Allow users to view and edit collected data
• Customizing applications to different situations without recompiling
• Collecting information from various sensing devices
• Usage of cheaper technologies (e.g., paper, SMS)
ODK-X Design Improvements

• Improve data management on mobile device
  – Better ability to modify previously collected data
  – Custom data views on mobile device (no PC)

• Use a row as the basic data structure to move and use across applications and client devices
  – Flat data structures (eliminate xml)
  – Synchronize data between devices and cloud
  – Data should be easily exportable to common formats

• Favor runtime languages
  – Easier to deploy customizations (no recompiling)
  – Easier for individuals with limited programming experience

• Increase diversity of input types
  – Enable new data input methods from built-in and external sensors
  – Reduce human data entry

Based on feedback from ODK users
Problem ODK is solving

• Many existing mobile frameworks are generally aimed at developers or users with significant technical skills and/or financial resources, making it difficult for organizations in resource-constrained communities to adapt to context dependent field deployments.

• Often mobile apps & frameworks assume connectivity or only short amount of time offline
TENSION: Generic vs. Customized

- Domain-specific/customized tools can be inflexible
  - Encourages the proverbial “re-inventing wheel”
  - Tool often cannot be reused in another similar domain
  - Keeps data siloed

- Users & Developers often find custom solutions easier
  - Can be modified to do exactly what the user wants
  - Developers can optimize performance and workflows
TENSION: Monolithic vs. Modular

Versus


Modularity for Open Source Ecosystem

- **Make tools highly modular and customizable**
  - Enables organizations to compose tools that are appropriate for their deployments
- **Keep design modularized so others can reuse code**
  - People can start with the basics already
  - Enables customization for specific use cases
  - Apache 2 License
- **Simplifies parallel development**
  - Avoid having to deliver everything at once
- **Companies can add features and create value**
Different Constraints Exist

• Until universal connectivity is a reality intermediate software solutions are needed

• Various ICTD research projects focus on extending Internet infrastructure
  – e.g., long distance WiFi, village base stations, mesh networks
Different Constraints Exist

- Until universal connectivity is a reality intermediate software solutions are needed
- Various ICTD research projects focus on extending Internet infrastructure
  - e.g., long distance WiFi, village base stations, mesh networks

Software Approach:
- Create customizable frameworks that are designed for disconnected operation
- Adjustable frameworks that leverage heterogeneous connectivity and adjust to changing networking conditions based on a deployment requirements.
EXAMPLE: Problematic Mobile Developer Paradigms/Assumptions

- **Uniform Data** – Existing transmission/routing paradigms assume inherent data qualities are sufficient to make decisions

- **Single-Task Mobile App** – Apps for mobiles are designed differently than PC utility programs (e.g., MS office)
  - Compete for resources

- **Similar Transmission Cost** – Assumption that transmission costs are similar everywhere (e.g., TCP/IP costs same everywhere)
Perspectives

• Single concept of “Application Layer” not sufficient
• Split to create Deployment & Application
• Examine from 3 perspectives
  – Software Application
  – Application Deployment
  – Platform

- Deployment
  - No software support
  - No platform support

- ODK X
  - User not involved

- Platform

- Software Application
Perspectives

- Single concept of “Application Layer” not sufficient
- Split to create Deployment & Application
- Examine from 3 perspectives
  - Software Application
  - Application Deployment
  - Platform

Recognized 4 roles:
- End-Users
- Deployment Architect
- Programmers
- ODK Framework Developers
• GOAL: Enable ‘Development Architects’ to adapt ODK to their deployment contexts by configuring multiple reusable frameworks
ODK 1.0 Architecture

CLOUD

AGGREGATE

Database

Fusion Tables

CSV

KML

MOBILE DEVICE

COLLECT

File System

FILES
ODK 2.0 Architecture

CLOUD

Sync-Endpoint

Database

CSV

MOBILE DEVICE

SCAN

Survey

Tables

Database

File System

Submit

Sensors
Bi-directional Transmission & Update
Data Synchronization
ODK Frameworks

• Collect
  – XForm-based mobile client for data collection, verification, and workflow
• XLSForm
  – Spreadsheet-based form authoring tool
• Survey
  – Mobile client framework for collecting rich data using complex workflows
• XLSXConverter
  – Spreadsheet-based form authoring tool
• Sensors
  – Framework to enable organizations to connect external sensors/hardware
• Submit
  – Framework to enable organizations to optimize transmission
• Tables
  – Framework to enable organizations to view and curate data on disconnected device
• Scan
  – Framework to enable organizations to bridge paper to digital (outside my PhD work).
ODK Frameworks

- **Collect**
  - XForm-based mobile client for data collection, verification, and workflow

- **XLSForm**
  - Spreadsheet-based form authoring tool

- **Survey**
  - Mobile client framework for collecting rich data using complex workflows

- **XLSXConverter**
  - Spreadsheet-based form authoring tool

- **Sensors**
  - Framework to enable organizations to connect external sensors/hardware

- **Submit**
  - Framework to enable organizations to optimize transmission

- **Tables**
  - Framework to enable organizations to view and curate data on disconnected device

- **Scan**
  - Framework to enable organizations to bridge paper to digital (*outside my PhD work*).
Framework Requirements

• Enables organizations to leverage mobile devices to build *customizable information systems*.

• Isolates the *user-configurable portion* of the framework from the *reusable system components*.
  – Create abstractions that are flexible and adaptable enough to support many different types of workflows from different subject domains.

• Facilitates the *integration of new capabilities* into the framework.
  – *Integrating new sources of information* (e.g., surveys, sensors) should be as simple as adding configuration files and data-handling routines to the framework.
ODK-X: HTML on the Device

- Highly Customizable
- Dynamic Graphs
- Watermarks
- Access Server Content (AJAX)

Only WebDev Skills

Redesign
ODK-X Architecture

Mobile Device

Cloud Hosted Service

OR

Local Server

SCAN

TABLES

SURVEY

SENSORS

Cloud Hosted Service

Local Server

Sync Service

Database Service

Web Server/File Service

SUBMIT
ODK-X Architecture

Survey Framework:
Framework for collecting data with verification using arbitrary workflows

Mobile Device

Cloud Hosted Service
OR
Local Server

SCAN
TABLES
SURVEY

SENSORS

Database Service
Web Server/File Service

Sync Service

SUBMIT
ODK-X Survey

• Mobile framework for:
  – Collecting strongly typed data
  – Directed navigation
  – Rendering complex workflows

• Different approach than ODK Collect (1.x)
  – Easier customizations
  – Easier branching and workflows
  – Easier to access databases

• Example: Pneumonia Detection (Ghana & India)

ODK-X Architecture

Tables Framework:
Framework to enable viewing and curating data on a disconnected device

Cloud Hosted Service
OR
Local Server

Mobile Device

SCAN

TABLES

SURVEY

SENSORS

Cloud Discovery
Sensor Manager
Database
Manager

USB
Manager

Bluetooth
Manager

Sync Service

Database Service

Web Server/
File Service

SUBMIT

Cloud Hosted Service

Local Server
ODK-X Tables

- Visualization of underlying database
- Provide users different looks at data
- User interface customizable in HTML/JavaScript
- **Example:** Cold chain monitoring

ODK 2.0 Architecture

Sensors Framework:
Framework to enable organizations to connect external sensors/hardware

Cloud Hosted Service

OR

Local Server

Scan

Tables

Survey

Services

Sync Service

Database Service

Web Server/ File Service

Submit

Mobile Device

SENSORS
## Sensor Diversity

- **Sensors differ by**
  - Communication Channel (e.g. Bluetooth, USB, NFC)
  - Type of data collected
  - Configuration

- **Four representative applications**

<table>
<thead>
<tr>
<th>APPLICATION</th>
<th>COMMUNICATION</th>
<th>CONFIGURATION</th>
<th>DATA STYLE</th>
<th>SENSOR RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medical (Heart rate monitor)</td>
<td>Bluetooth</td>
<td>Calibrate</td>
<td>Real-Time Single Reading</td>
<td>~ 1.5 packets / sec</td>
</tr>
<tr>
<td>Milk Bank (Temperature sensing for milk pasteurization quality control)</td>
<td>USB</td>
<td>Sampling Rate</td>
<td>Real-Time Time-Series</td>
<td>~ 1 packet / sec</td>
</tr>
<tr>
<td>Vaccine (Temperature monitoring and alerts for vaccine quality control)</td>
<td>USB</td>
<td>Alerts Sampling Rate Snapshot Size</td>
<td>Snapshot Time-Series</td>
<td>~ 1 packet / sec</td>
</tr>
<tr>
<td>Water Time (Sensor to record movement intervals)</td>
<td>Bluetooth</td>
<td>Identifier Calibrate</td>
<td>Historical Time-Series</td>
<td>~ 50 packets / sec</td>
</tr>
</tbody>
</table>
End-User View

Two Android “Apps”
- User Application App
- Framework App

Hardware Pieces
A. Android Phone
B. Arduino Board (USB Bridge)
C. Temp/Current Sensor (I2C)
D. Heart Rate Sensor (Bluetooth)
ODK-X Sensors

**GOAL:** Enable a market of *reusable application components* that can be easily integrated by non-technical users to create and deploy sensing applications.

**Make it EASY to:** Connect sensors of any type
- Deployable to *any* Android consumer device
- Drivers should be easy to add and auto-upgrade (e.g. Windows Update)

**SOLUTION:** *Framework with “User-level” sensor drivers*
- No operating system modifications
- Allows convenient reuse between applications
- Distribution through existing app store model
- Create a single sensor interface
- Separate concerns (Drivers vs Framework)
- Framework handles all STATE MANAGEMENT

---

Separation of Concerns

• Application Developer
  – implements top-level user applications

• Driver Developer
  – creates sensor-specific processing and control modules

• Framework Developer
  – Absorbs as much work as possible
  – provides the BT/USB HW Setup, Threads, Sockets, Buffers, etc
ODK-X Architecture

Submit Framework:
Framework to enable organizations to optimize data transmission

Mobile Device

Cloud Hosted Service

OR

Local Server

SCAN

TABLES

SURVEY

SENSORS

SERVICES

Sync Service

Database Service

Web Server/File Service

SUBMIT
ODK-X Submit

• **Flexible data transmission framework**
  – Enables *Deployment Architects* to optimize data transmission to contexts
  – Common network abstractions
    • *Unify* many different data transmission channels
    • *Hide* peer-to-peer channel details
    • *Simplify* app development (can use multiple channels to send data)
      – Avoids additional software development by leveraging submit framework
  – Expose a *simplified* set of data characteristics
  – Optimize communication based on contextual requirements.

Peer-to-Peer Transfer

- Facilitate network connectivity in remote regions using peer-to-peer communication

- GOAL: Enable ‘Deployment Architects’ to adjust P2P to contextual requirements
  - Examine 5 different Android P2P transfer methods
ODK-X TOOL SUITE

To validate the derived requirements, ODK-X has been tested in 6 case studies.

<table>
<thead>
<tr>
<th>CASE STUDY ODK-X FEATURE REQUIREMENT SUMMARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Childhood Pneumonia</td>
</tr>
<tr>
<td>----------------------------</td>
</tr>
<tr>
<td>Complex/Non-Linear Workflows</td>
</tr>
<tr>
<td>Link Longitudinal Data to Collected Data</td>
</tr>
<tr>
<td>Data Security and Use Permissions</td>
</tr>
<tr>
<td>Reuse of Data Fields Across Forms</td>
</tr>
<tr>
<td>Bidirectional Synchronization</td>
</tr>
<tr>
<td>Customizable Form Presentation</td>
</tr>
<tr>
<td>Custom JavaScript Apps</td>
</tr>
<tr>
<td>Sensor Integration</td>
</tr>
<tr>
<td>Paper Digitization</td>
</tr>
<tr>
<td>Custom Data Type Update Multiple Fields in a Single User Action</td>
</tr>
</tbody>
</table>
OVERVIEW

RC² Relief improves relief cycle processes from emergency assessment through to distribution, reporting, and monitoring.

Adaptable for a variety of humanitarian contexts including cash transfer programming and the distribution of relief items.
OVERVIEW

RC² Relief Tool is applicable to many scenarios:

- Basic delivery using a re-usable barcode (e.g., serving a meal)
- Registration of “Beneficiary Entities” with criteria based on delivery of relief items.
- Multiple individuals can be grouped in “Beneficiary Entities” (e.g., households)
- Beneficiary follow-up for long-term rebuilding programs
## THE ADVANTAGES

| **Streamlines the collection and use of information through bidirectional synchronization of data.** |
| --- | --- |
| **User profiles aligned with the structure of the organization control access to increase data security and privacy for the assisted people.** |
| **The mobile application can store data on past assistance, without the need for an Internet connection.** |
| **The ability to custom workflows based on field conditions reduces errors and data duplication.** |
### THE ADVANTAGES

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- **Improves efficiency of field workers**, allowing more people to be reached in situations with limited resources.

- Facilitates accountability through **improved record-keeping**.

- Adaptable for a variety of humanitarian contexts including **cash transfer** programming and the **distribution of relief items**.
ODK-X TOOL SUITE

It is an open-source suite of tools that helps organizations create, administer, and manage mobile data collection solutions.

ODK-X had an iterative requirements gathering process:

- Surveys
- Pilot deployments in 18+ countries by a variety of organizations,
- The ODK-X tool suite went through a significant redesign from the original ODK-X vision
DESIGNED FOR FLEXIBILITY

A design priority was making the RC² Relief tool flexible so National Societies could customize RC² Relief based on their available resources.

Modular design for changes to “disaster distribution pipelines.”
- Enables each National Society to customize or share their forms and templates
- Multiple output stages as simple CSVs for human customization in the field
- Enables innovation as small changes to based on the needs of the disaster response or other field conditions can easily introduced
- Future modules can incorporate varying business logic while still remaining compatible with the rest of the application
Challenges/Lessons Learned

• Challenges involved in designing six mobile frameworks to work together seamlessly on the mobile device
  – *Part of modularity and open source ecosystem goal*
  – *Goal frameworks can work independently or together to make a more complex system*

LESSON: Schema helped

- ODK 2.0 is **database-centric** instead of **file-centric**
  - DB rows are the basic unit of storage of ODK 2.0
  - View definitions and settings stored in files (not data)

- **Helps Deployment Architects understand how to use the 5 frameworks together because they are in control of the common DB schema for all tools/frameworks**
  - Also helps with deployment issues
    - Pushes *Deployment Architects* to avoid collecting data and not understanding how to process the data
    - *Deployment Architects* can avoid conflicting disconnected updates between users in rows by separating data tables
    - Can easily separate out important data to transmit more quickly rather than data that is less important
Roadblocks are not what you expect!
Contribution Summary

- Open Data Kit (ODK) enables organizations to create domain independent mobile information management solutions by providing customizable mobile frameworks that
  - are designed to adapt to challenged mobile networking conditions.
  - create new abstractions that are usable by non-programmers with limited technical expertise (e.g., Deployment Architects)
  - are modularized to enable interoperability of tools that can be used together or separately to simplify customization
Questions?

Thank you for your attention

http://www.opendatakit.org

http://change.washington.edu