Journal of
Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7766123

OPEN DATA KIT (ODK) FOR RELIABLE AND LONG-TERM DATA COLLECTING FOR DEVELOPING REGIONS

[1]Dr. KD Gupta, [2]Punit Soni
Head & Ph.D. Guide, Research Scholar
Apex University Jaipur

ABSTRACT

The Open Data Kit (ODK) is a free, open-source toolkit that enables businesses to create information services tailored to certain applications for usage in environments with limited resources. One of the best data gathering tools is ODK, which is used by a wide range of organisations in several nations. This document describes how we recently redesigned the ODK system architecture in response to feedback from users and developers. To make customizations simpler for people with little programming experience, the design principles for ODK specifically emphasise the following: 1. Prefer runtime languages over compile-time languages; 2. Implement fundamental data structures as single rows in a table of data; 3. Store that data in a database accessible across applications and client devices; and 4. Increase the diversity of input types by enabling new data input methods from sensors.

We highlight how these ideas have influenced the development of numerous new tools that attempt to enhance the toolkit, broaden its scope of uses, and make it more individualised for users as well as the improvement of the current ODK products.

Keywords: A large Data collection Smart Phone (Android phone) Software Technology. Data collection · Mobile data, Mobile device · Open Data Kit · ODK · ODK 2 · ODK-X · Tool · Architecture, smartphones, mobile databases, spreadsheets, data tables, Bar Code.

1. INTRODUCTION

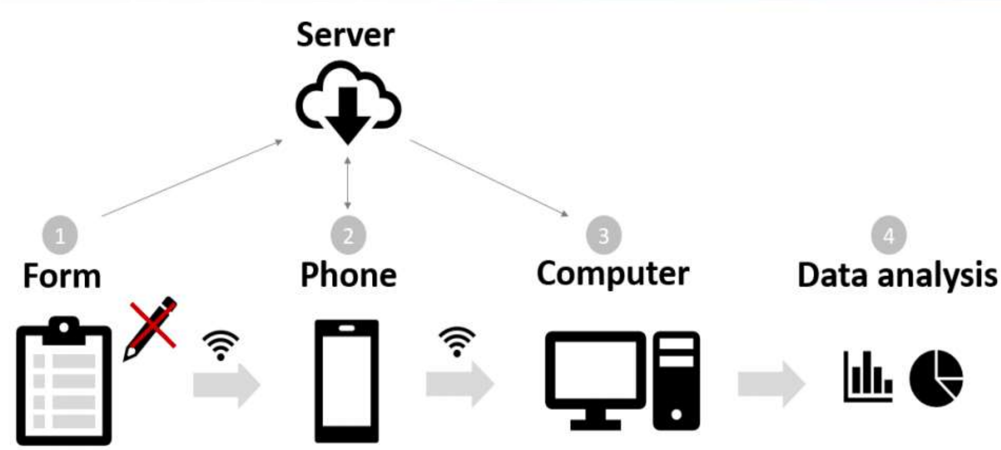
The preferred platform for implementing data collecting and information services in the developing world is swiftly evolving smart android phones. Due to their portability, improved independence from the power infrastructure, ability to connect to the internet through cellular networks, and reasonably easy user interfaces enabling well-targeted applications for many areas, they have swiftly surpassed desktop and laptop computers. In effect, poor nations are employing smartphones and tablets to complete jobs that were previously carried out on larger computers rather than the desktop and laptop phases of computing growth. In line with this growth, cloud services give several businesses the flexibility to quickly rent data storage space and expand hosting capabilities as necessary, locally, or globally. When we started the Open Data Kit (ODK) project at the University of Washington, we were aware of two trends: 1) competent client devices with sophisticated user interfaces and 2) cloud-based scalable data gathering, computation, and visualisation services.

By developing ODK, we want to provide organisations with constrained financial and technical resources with a modular, evolvable toolset for developing data collecting and dissemination services. We selected Android as our development platform because of its adaptable inter-process communication techniques, which allowed us to use pre-existing apps for functions like taking photos, scanning barcodes, and locating our location without having to create them from scratch. This sped up development. The full set of modular tools is referred to as ODK. Each of the three main tools in the ODK suite—Build, Collect, and Aggregate—has been given a name that reflects what it does.

DESIGN OF ODK

The development and growth of ODK are based on four fundamental design concepts, which we are implementing across all of the tools (although they do not precisely equate 1-to-1 with the four areas of refinement stated at the beginning of Section 2): 1. UI components should, where feasible, be created using a runtime language rather than a compile time language to make adjustments simpler for users with basic programming knowledge. Data should be stored in a database that can be shared across devices and easily extracted to a variety of common data formats; 2. the basic data structures should be easily expressible in a single row; 3. nested structures should be avoided when data is in the display, transmission, or storage states; and 4. new sensors, data input methods, and data types should be simple to incorporate into the data collection pipeline by people with limited technical experience.

How the ODK Cycle Operates: -



Data Management on Mobile Devices

Logistics management, public health, and environmental monitoring are just a few applications that rely on previously gathered data. These apps frequently demand that users return and refer to previously obtained data to confirm and modify needs. Revision of data from the previously completed surveys was not supported, though, under the previous ODK design. To fill out new surveys, more and more people want to be able to use all their responses from earlier ones (e.g., not re-entering patient demographics for a follow-up visit when that data was already collected in the original registration form).

Improved Input Methods

ODK reduces the vast manual data transcription from sensors into surveys by making it possible to attach external sensors to mobile devices. By hiding complexities, including managing communication channels and sensor state and data buffering and threading, sensors framework simplifies the code needed to access a sensor management of communication channels and sensor state as well as data buffering and threading. The sensors framework simplifies the code needed to access a sensor. In addition to accepting and processing input from different sensors, the continued use of paper forms for data collection in resource-constrained environments made it significant that we facilitate efficient data entry from paper forms. Many paper forms used by organizations for data collection contain a mixture of data types, including handwritten text, numbers, checkboxes, and multiple-choice answers. While few data types, such as handwritten text, require a person to transcribe the data manually, others, like checkboxes or bubbles, can be analysed and interpreted automatically.

Data Management in the Cloud

Users that are less technically skilled face substantial obstacles when trying to take use of cloud computing. We created Aggregate, an autoconfiguring, ready-to-deploy server, to simplify the distribution of forms to mobile devices, the retrieval of data from devices, and the storage and management of data. Aggregate handles collected data, provides interfaces to export the aggregated data into standard formats (e.g., CSV, KML, JSON), and allows users to publish data to online services (e.g., Google Spreadsheet or Fusion Tables). Aggregate is a configurable generic data storage service that runs on a user's choice of computing platform (cloud-based or private server). Aggregate is installed to the Google App Engine hosting service that enables unskilled users and less-capable IT organizations to maintain a highly available and scalable service. However, many of our users have data locality and security concerns, either because the data cannot legally leave the country of origin or because the data may contain sensitive perceptible information or be high-risk or high-value data. For these users, App Engine may not be suitable. Aggregate can run within a Java web container (e.g., Tomcat) using a MySQL or PostgreSQL datastore. HTTPS connections between client devices and the server are often essential for communication security. However, many businesses lack the funds or best knowledge necessary to put SSL certificates on their servers. We utilise Digest AUTH to offer user authentication and data security via HTTP interactions, and we encrypt form data using asymmetric public keys before sending it to the cloud. Asymmetric public-key encryption enables certain enterprises to continue using the App Engine cloud hosting service despite strict data security requirements by storing form data in encrypted form on the server. In this

instance, users download the encrypted data to a computer, where they may then use a private key and a locally installed programme called ODK Briefcase to decrypt the data.
In ODK 1.0, the communications flow comes from one single direction. The blank forms flow from the cloud service (Aggregate) to mobile devices, and data from the filled-in forms flows back to the cloud service and then moves to remote services or towards the file exports. Collected data can be deleted but is otherwise immutable and provides a store of record. Data is stored (aggregated) in the cloud, where simple curation and data visualization tools are provided. Aggregate bridges the gaps between mobile data collection tools and sophisticated data analysis software. It can derive complex results by providing many forms of data export.
Because connectivity is frequently unpredictable and patchy and because the businesses want to keep data transmission costs under control, data submission is typically started by the user. The team created a tool called Submit that controlled the data transfer to make better use of the potential inconsistent connectivity and to increase data timeliness (both on the mobile device and while publishing data to the peers). Organizations can establish criteria like data priority, data relevance, timeframes, and the cost of the transport mediums with submit. Submit factors in the device's connectivity history and intelligently uses the connectivity available (e.g., SMS, GPRS/3G, Wi-Fi) to create a priority routing system that improves data timeliness in the intermittent and expensive connectivity of the developing world. Connectivity history is a significant factor in routing decisions since there may be certain times of day when the device is within range of a Wi-Fi base station. Alternatively, depending on the data priority and the costs of other connectivity options, it may make sense for the data to be stored locally until the user returns to Wi-Fi connectivity.

FUTURE WORK & CONCLUSION

The original design of ODK assumed that a system administrator would have access to a computer to initially set up and administer the system, including designing forms and setting up data storage facilities. However, in many rural environments, computers are rarely available, which limits the adoption of ODK in these settings. To increase the accessibility of ODK in rural locations, it is good to create a system that could be entirely set up and administered on a mobile device. While ODK gives users certain tools for creating information systems for mobile devices (such as custom inquiry widgets and database architecture using Tables), it does not eliminate dependency on PCs because users cannot set up their cloud service or create a device driver. To create a system that can be installed and used on a mobile device, more effort is needed. The new ODK design concentrates on a core set of capabilities that let users utilise mobile computing platforms to create more dynamic collaborative information systems in the field rather than simply using mobile devices as basic input devices. We anticipate that the software's increased functionality and modifications to its architecture will provide a new set of research possibilities and difficulties that we intend to investigate.
Conflict of Interest: This work was supported by Apex university Jaipur I am grateful to Dr. Kd Gupta of Apex university Jaipur without his guidance, this work would not have been possible.
Funding Source: "There is no funding source for this study"
Acknowledgement: I author declare that there are no competing interests.

References

1. forum.getodk.org/t/odk-aggregate-v2-0/17706.
2. W. Brunette, R. Sodt, R. Chaudhri, M. Goel, M. Falcone, J. Van Orden, G. Borriello, "Open Data Kit Sensors: a Sensor Integration Framework for Android at the Application-level," Proc. 10th Intl. Conf. on Mobile Systems, Applications, & Services (Mobisys), 2012.
3. OpenDataKit. [Online].; 2017. Available from: HYPERLINK "https://opendatakit.org/" https://opendatakit.org/ .
4. B. DeRenzi, G. Borriello, J. Jackson, V. S. Kumar, T. S. Parikh, P. Virk, and N. Lesh, "Mobile Phone Tools for Field-Based Health Care Workers in LowIncome Countries," Mount Sinai Journal of Medicine, vol 78, no 3, 2011.
5. Floyd J. Fowler J, Mangoine TW. Standardized Survey Interviewing: Minimizing Interviewer - Related Error: SAGE Publications; 1990.
6. Froehlich, M. Y. Chen, S. Consolvo, B. Harrison, and J. A. Landay, "MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones," Proc. 5th Intl .Conf. on Mobile Systems, Applications and Services (Mobisys), 2007.
7. C. Hartung, A. Lerer, Y. Anokwa, C. Tseng, W. Brunette, and G. Borriello, "Open Data Kit: tools to build information services for developing regions," Proc. 4th ACM/IEEE Intl. Conf. on Information and Communication Technologies and Development (ICTD), 2010.
8. Y. Hong, H. K. Worden, and G. Borriello, "ODK Tables: data organization and information services on a smartphone," Proc. 5th ACM Workshop on Networked Systems for Developing Regions, 2011.
9. B. Kaufmann and L. Buechley, "Amarino: a Toolkit for the Rapid Prototyping of Mobile Ubiquitous Computing," Proc. 12th Intl. Conf. on Human Computer Interaction with Mobile Devices and Services, 2010.
10. Y.S. Kuo, S. Verma, T. Schmid, and P. Dutta, "Hijacking Power and Bandwidth from the Mobile Phone's Audio Interface," Proc. 1st ACM Symp on Computing for Development (DEV), 2010.
11. F. X. Lin, Z. Wang, R. LiKamWa, and L. Zhong, "Reflex: Using Low-power Processors in Smartphones without Knowing Them," Proc. 17th Intl .Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2012.
12. T. S. Parikh and E. D. Lazowska, "Designing an Architecture for Delivering Mobile Information Services to the Rural Developing World," Proc. 15th Intl Conf on World Wide Web (WWW), 2006
13. S. Van Wart, K. J. Tsai, and T. Parikh, "LocalGround: a Paper-based Toolkit for Documenting Local Geo-spatial Knowledge," Proc. 1st ACM Symposium on Computing for Development (DEV), 2010.
14. L. Wei-Chih, M. Tierney, J. Chen, F. Kazi, A. Hubard, J. G. Pasquel, L. Subramanian, and B. Rao, "UjU: SMS-based applications made easy," Proc 1st ACM Symp. on Computing for Development (DEV), 2010.

15. Groves RM, Couper MP. Nonresponse in Household Interview Surveys: John WIley and Sons; 1998.

16. Vardigam M, Granda P, Hoelter L. Documenting Survey Data Across the Life Cycle. In Wolf C, Joye D, W.Smith T, Chi-Fu Y. The SAGE Handbook of Survey Methodology.: SAGE reference; 2016. p. 443 - 459.

17. Android Accessory Development Kit. http://developer.android. com/tools/adk/index.html. [Accessed: 11-Oct-2012].

18. Open_Data_Kit_20_Expanding_and_refining_information_services_for_developing_regions. [Accessed: Feb-2013].