Mobile Field Data Collection for Post Bushfire Analysis and African Farmers

Bradley Lane, Nicholas J. Car, Justin Leonard, Felix Lipkin, Anders Siggins

To cite this version:

Bradley Lane, Nicholas J. Car, Justin Leonard, Felix Lipkin, Anders Siggins. Mobile Field Data Collection for Post Bushfire Analysis and African Farmers. 11th International Symposium on Environmental Software Systems (ISESS), Mar 2015, Melbourne, Australia. pp.160-168, ⟨10.1007/978-3-319-15994-2_15⟩. ⟨hal-01328545⟩

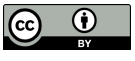
HAL Id: hal-01328545

https://inria.hal.science/hal-01328545v1

Submitted on 8 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Mobile Field Data Collection for Post Bushfire Analysis and African Farmers

Bradley Lane[1], Nicholas Car[1], Justin Leonard[1], Felix Lipkin[1], and Anders Siggins[1]

[1]Land & Water Flagship: CSIRO, Highett & Clayton Vic and Brisbane Qld, Australia

bradley.lane@csiro.au

Abstract. In recent years CSIRO has been trialling field data collection using mobile devices such as phones and tablets. Two recent tools that have been developed by CSIRO are the CSIRO Surveyor (Post Bushfire House Surveyor) and DroidFarmer. Challenges tackled include mapping field documents to mobile data through QR (Quick Response) codes, rapid input of survey data, accurate capture of GPS locations and offline operation. Throughout this paper we detail the design choices made for these systems. We give details of how well field data collection was performed and discuss our planned future developments in this space.

Keywords: DroidFarmer · CSIRO Surveyor · AusAID · Bushfire · Africa · Farmer · GIS · Questionnaire · Android · MongoDB · SQL Server

1 Introduction

CSIRO Surveyor and Droid Farmer applications (apps) were designed to fulfil a need to easily and efficiently collect field data. Each tool faced different challenges due to differing design requirements. The key issues for Droid Farmer were linking field documents to app-collected data, getting accurate GPS readings and enabling offline data recording (i.e. local phone storage) for areas with poor network coverage. CSIRO Surveyor had to handle issues of efficiency in GIS and survey data entry, map caching for offline use and the possibility of communications infrastructure breakdown.

Throughout this paper we explain how we attempted to resolve these issues through mobile application system design and software architectures.

2 CSIRO Surveyor

CSIRO Surveyor was initially developed for the New South Wales Rural Fire Service (RFS) to enable detailed data capture in areas recently affected by bushfire. After a bushfire front passes an area and it is deemed safe, trained survey teams with tablets carrying the Surveyor application are sent into the field. The app delivers two primary functions: firstly, the detailed marking of houses, trees, water tanks and other types of

© Springer-Verlag Berlin Heidelberg 2015

objects using a Graphical Information System (GIS) user interface (see Fig. 2) and secondly, a novel Survey Questionnaire Response System (SQRS, see Fig. 3). Along with the main app a cut-down version was developed for Rapid Impact Assessment, which provided a means to easily visually track bushfire-affected houses via a central desktop GIS dashboard.

The data collected was used for public reporting of statistics and then post bushfire analysis. Analysis of the affected areas gives information on how to reduce the impact of bushfires in the future.

2.1 Basic System Design

The basic system design for CSIRO Surveyor is displayed in Fig 1.

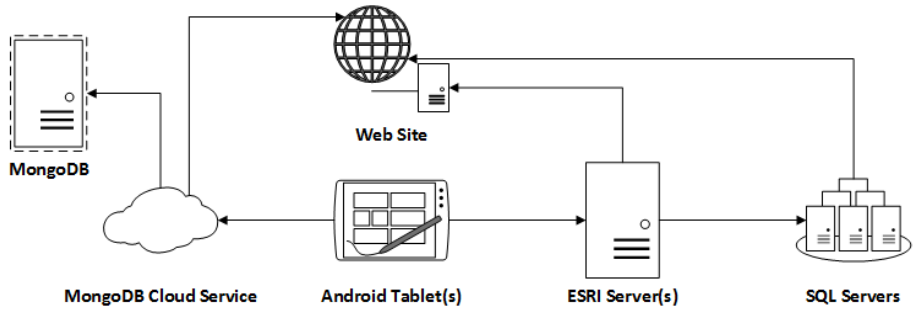


Fig. 1**.** The main technologies used were ESRI GIS Servers, SQL Databases (for ESRI data and Questionnaires), Android Devices, ASP.NET Web Sites and MongoDB Cloud Services.

[4] discusses a similar Android data collection system and points to the importance of ease of input, consideration of the availability of internet connectivity and data visualisation. We address these considerations below.

GPS Touch Input

The GIS interface (see Fig. 2) was developed using Android and ESRI [1] APIs. Maps are loaded before travelling to a site. Once on location, the user marks houses, trees, water tanks and other entities of interest regarding bushfires by first touching the entity type on the right and then marking its location on the screen. After marking the entity locations the questionnaire activity starts.

There were three different methods of marking entity locations on the screen depending on the type of entity. To record a polygon entity such as a house the user marked each point and then pressed the starting point again to complete the shape. For a single point entity the user just marked the desired location once. Finally for multi-point entities like trees the user marked each position with a screen tap for each point and then completed the entry process by tapping the final point a second time.

Questionnaires

The tablet-optimized Survey Quick Response System (SQRS) was developed using the standard Android API. Questionnaires were structured in a hierarchical fashion using a separate web tool and then uploaded to a central MongoDB cloud-base store [2] on creation; editors were also able to update the cloud store questionnaires. Tablets then communicated with the cloud store allowing them to automatically update their local copies of the questionnaires. Questions could be 'removed' by disabling/hiding them. No questions were deleted from the cloud store to prevent older responses being orphaned (no question left to link to an answer).

Fig. 2. The above image shows the GIS input interface.

Questionnaire Construction

Questionnaires were built by creating questions with appropriate responses and adding them to a hierarchy which ensured only necessary information was collected when certain question's answer invalidated other questions. Questions responses could be either multiple choice or numeric, with individual responses being linked to any number of further questions in the hierarchy.

Multiple choice responses were the simplest to implement as new questions could be easily linked to individual responses. Linking numeric responses to new questions was achieved using value ranges, for example, when a user enters value in range of 0 to 20 we could ask questions x, y and z.

Questionnaire Input

Due to the hierarchical structure of the questionnaires users could complete input actions very quickly. For multiple choice questions, a single tap on a response immediately moved one to the next appropriate question. For numeric answers, the user had to enter a value before tapping a button to continue.

When a mistake was made, the user could press a 'Back' button to backtrack through the questions. The questions followed a specific track, with questions being asked until the end of a branch with hierarchy recursion used to find the next relevant question. A summary of questions and responses is given on the right side of the SQRS interface in Fig. 3.

Some questions required the user to enter information regarding a specific face or side of a building. To make this process easier for the user we dynamically labeled them for the user as A, B, C and D (see Fig. 3) then referred to these specific labels in the questionnaires.

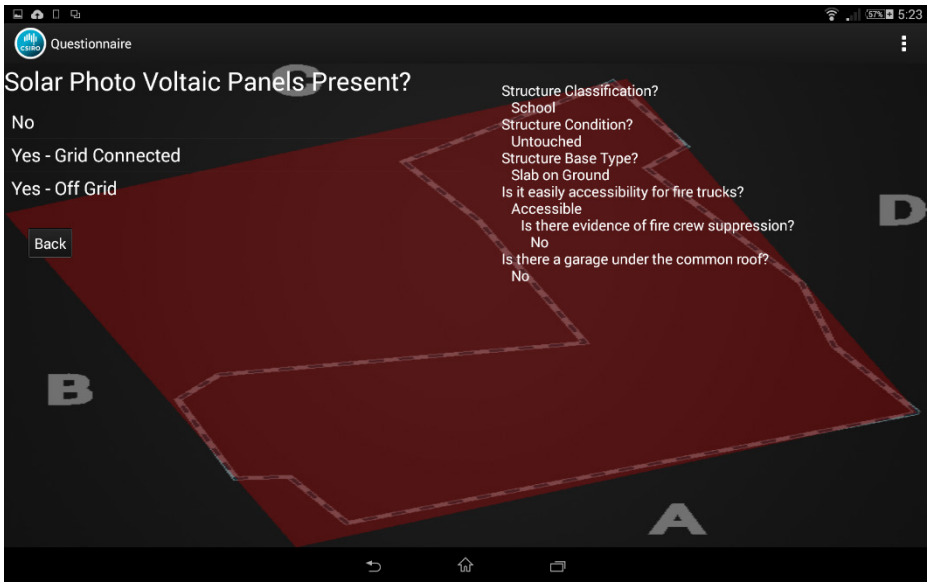


Fig. 3**.** The above image shows the SQRS input interface.

2.2 Consideration of Infrastructure Breakdown

To cater for mobile phone network infrastructure breakdown, the CSIRO Surveyor and RIA tools incorporated offline capabilities. This was implemented using a SQLite database to cache data that had not been sent to the cloud store. When the user of a tablet caching data later moved into network connectivity, the cached data could then be transmitted. Since data could be cached, phone network infrastructure state does not affect the effectiveness of the app. Results could not be displayed on the desktop

GIS dashboard until successful data upload.

2.3 Data Integrity and Cross Checking

For comparison and data integrity purposes, two copies of the questionnaire results were transmitted; one to the ESRI Database and one to MongoDB. This was very helpful in both ensuring that results were successfully transmitted to both systems and that the results matched. Data was transmitted with a globally unique identifier (GUID) and a time stamp for quick matching.

2.4 Usage Statistics

Between October 20th and 30th 2013, 4,715 questionnaires were submitted from one bushfire. These included 24 different types of questionnaires plus separate storage of other types of data such as photos. To collect the data from all affected houses approximately 12 teams with 2 people in each team were deployed into the field.

2.5 Planned Improvements

The current system is designed for use with bushfires but could easily be modified to support other disasters. To accomplish this, we intend to make certain sections of the system more generic. For example, the interface for selecting the type of entity to record (see Fig. 2) is partially hard coded. By making the screen layout and entities configurable, the application could easily support different disaster types.

3 DroidFarmer

DroidFarmer is an Android mobile phone application built to collect crop and spatial data for farm surveys in Africa. The app was designed using standard Android tools and built for compatibility with early Android devices. It was also designed to be very simple to operate in order to cater for users not familiar with Smartphones. It used offline storage with online sync to handle poor cellular/wireless internet availability in farm areas. GPS readings were taken for spatial objects such as trees and field areas and with users walking to locations and tapping a single screen button to record position. After transmission to a central server, recordings were displayed as map points and polygons on a web map interface.

This app is essentially a ‘mapping app’ as described by [5] however, unlike the proposition in that paper that such apps require potentially costly and time-consuming post-collection data analysis, such steps are not required where app users are the beneficiaries of the field data collection (e.g. farmers using the app to measure their field areas and results can be related on screen). Where scientists are the beneficiaries of aggregated field data, normal results processing is required.

The data collected had numerous uses. One example was the collection of soil moisture data from field instruments to assist farmers in choosing appropriate crops.

Another example was the recording of field size and position, which enabled farmers to order goods and services without being overcharged.

3.1 Basic System Design

The basic system design for DroidFarmer is shown in Fig 4.

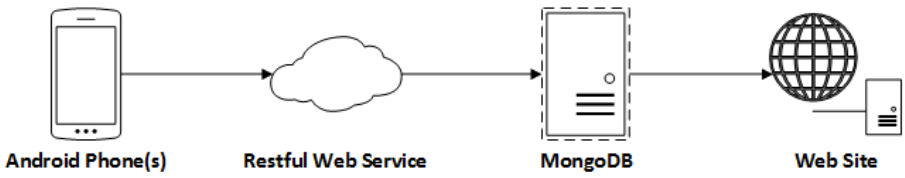


Fig. 4**.** Main technologies used were Android Devices, MongoDB RESTful Web Services and a Web Site.

Document Linking

One novel aspect of DroidFarmer was its linking of paper field records, readings from field equipment and phone gathered data through interpreting Quick Response (QR) codes and its function reuse.

Fig 5. The DroidFarmer logo and a QR Code

QR codes were printed on paper survey sheets and affixed to field instruments via stickers. This was designed to prevent data confusion and loss when data was recorded in multiple forms. The approach worked well with many thousands of complete bundles of field data being recorded and incomplete bundles instantly identifiable after data upload. This rapid understanding of data gathering completeness saved much time in deciding which fields to resurvey. Extensive input validation of phone input data and QR recording sequences was implemented which reduced data entry errors compared with paper-based recording. Several data gathering workflows were implemented that utilized the same GPS recording, QR code handling and form entry functions which are able to be used still further. The code scanning was done using a 3rd party tool called “Barcode Scanner” developed by ZXing Project [3]. Data visualisation was done via a website (see Fig. 6), where recorded data was marked on a map with pins; additional data and images could then be accessed by clicking on a pin.

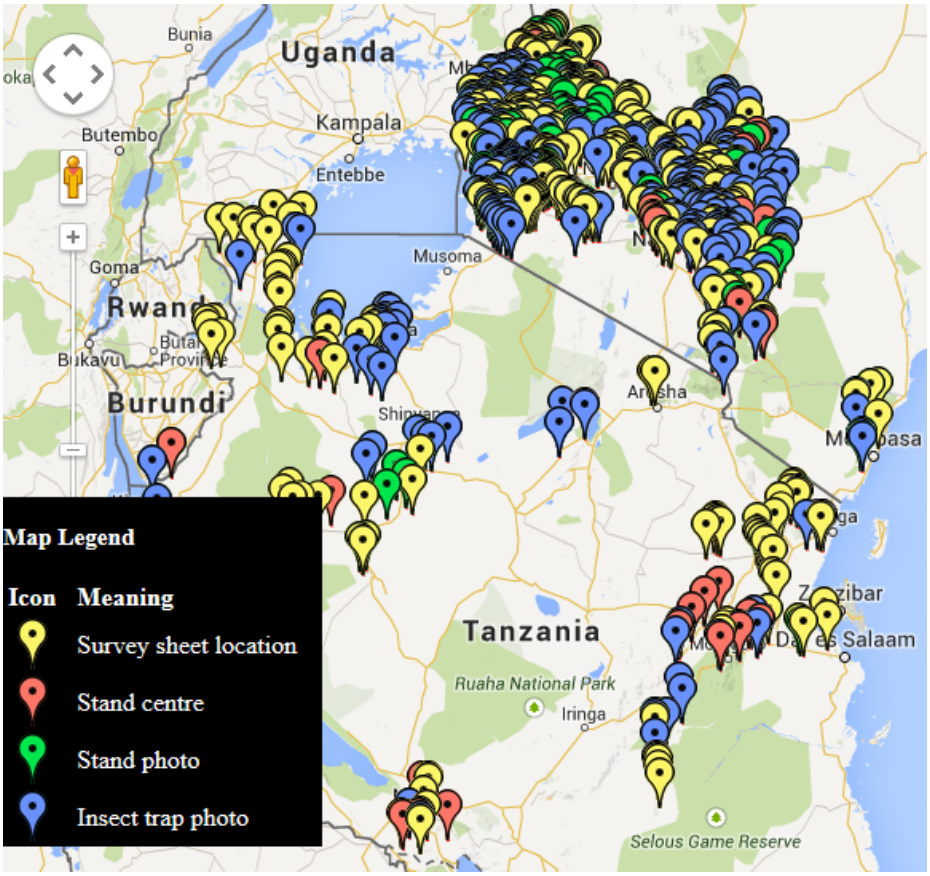


Fig. 6. A screenshot of a large number of DroidFarmer-reported survey locations in East Africa. Four different types of icons are shown for four different data types: the locations of survey sheet completions (in the field), the centres of stands of crops, photos of stands of crops and photos of insect traps.

Improving the accuracy of GPS readings

To aim for the best accuracy, we deployed multiple strategies based on observations from testing. Two key things we observed in testing were that GPS tracking took time to initiate and that accuracy improved the longer a user stayed in one spot.

To take advantage of this we started tracking locations as soon as the user entered the activity. Once started we continually polled for new readings with each new reading being added to a short stack. When a user requested a location be marked we would first look at the top reading, if it met our accuracy requirements it was used, if not we would check the stack for one that does, if none was found we took the average position.

Because the user needed to spend time entering data before taking a reading (not a coincidence), the device was normally in a good position to take the first reading. In either case the process was fast because the stack was continually updated the moment the activity was started.

Our goal was to have readings accurate to 10m or less. Using the above mentioned approaches we were generally able to achieve this target.

3.2 Dealing With a Low Tech Community

The overall level of smart phone technology was low in Africa. Due to this we needed to design the software to run on the lowest possible version of Android while taking into consideration our design needs.

3.3 Data Validation and Statistics

Most spatial data could be easily validated visually (GIS view) from the website upon the users return. At the time of writing there were several thousand submitted recordings across the various categories.

4 Controlling the User Experience

Use of the CSIRO Surveyor app was a controlled user experience as it was manually loaded onto tablets which were then handed to individuals and small groups after receiving training. DroidFarmer on the other hand was delivered to phones using the Google Play online app store meaning access to it was available to all. Scientists familiar with Smartphones wanting to capture data could do so easily, but farmers not familiar needed to be guided through the process.

5 Platform Specific vs Platform Independent

After considering various platform-specific vs platform-independent designs, we decided to go for a hybrid model. We utilised RESTful technologies whenever possible; but for devices we chose to be dependent on Android. This decision was made after careful consideration of the level of control we needed over device hardware.

6 Conclusions

Both DroidFarmer and CSIRO Surveyor were received well by their intended audience. Users appreciated the effort that was taken to meet their specific needs which we were able to achieve while retaining a large amount of application code reuse. DroidFarmer enabled, and continues to enable, farmers and researchers to collect data in Africa and have it instantly available for study world-wide. CSIRO Surveyor is continuing to enable the rapid collection of post disaster survey data. These systems

met their intended design use and are continuing to be developed and enhanced for future use.

References

1. Environmental Systems Research Institute, Inc. (2013) ArcGIS Runtime SDK for Android. Software Application. ESRI, Redlands, CA, USA. https://developers.arcgis.com/android/ Accessed 2014-10-30.
2. ObjectLabs, Corporation. (2014) MongoDB-as-a-Service. MongoLab, San Francisco, CA 94110, United States. https://mongolab.com/. Accessed 2014-10-30.
3. GitHub, Inc. (2014) ZXing Project (Barcode Scanner). https://github.com/zxing/zxing/. Accessed 2014-10-30.
4. van der Schaaf, H.; Rood, E. & Watson (2013) A Mobile Application for Reporting Disease Incidents. Environmental Software Systems. Fostering Information Sharing, K. Hřebíček, J.; Schimak, G.; Kubásek, M. & Rizzoli, A. (Eds.), Springer Berlin Heidelberg, 2013, 413, 188-195.
5. Aitkenhead, M.; Donnelly, D.; Coull, M. & Black, H. (2013) E-SMART: Environmental Sensing for Monitoring and Advising in Real-Time Environmental Software Systems. Fostering Information Sharing, K. Hřebíček, J.; Schimak, G.; Kubásek, M. & Rizzoli, A. (Eds.) Springer Berlin Heidelberg, 2013, 413, 129-142