Use Cases for Real Time Data in Agriculture

D. Buckmaster[1], J. Krogmeier[2], A. Ault[2], S. Noel[1], Y. Wang[2], Y. Zhang[2], A. Layton[2], A. Balmos[2]

[1] Agricultural and Biological Engineering, Purdue University, West Lafayette, IN
[2] Electrical and Computer Engineering, Purdue University, West Lafayette, IN

A paper from the Proceedings of the

14th International Conference on Precision Agriculture
June 24 – June 27, 2018
Montreal, Quebec, Canada

Abstract. *Agricultural data of many types (yield, weather, soil moisture, field operations, topography, etc.) comes in varied geospatial aggregation levels and time increments. For much of this data, consumption and utilization is not time sensitive. For other data elements, time is of the essence. We hypothesize that better quality data (for those later analyses) will also follow from real-time presentation and application of data for it is during the time that data is being collected that errors can be corrected and improvements of settings, operations, and protocol can be implemented.*

The objectives of this work were to develop open source real time data exchanges for the purposes of logistics, profitability, and data quality. The real time exchanges can require edge or cloud computing for analysis and visualization. The examples to be shared are open source contributions to the agriculture community related to: 1. Instant individual livestock records according to personal observation. 2. Presence/activity tracking to facilitate shared record keeping for improved logistics of personnel and machinery, 3. Auto-generation, analysis, and display of machine-based data in cropping systems to facilitate records and machinery and labor efficiency metrics "by the field".

The authors are solely responsible for the content of this paper, which is not a refereed publication.. Citation of this work should state that it is from the Proceedings of the 14th International Conference on Precision Agriculture. EXAMPLE: Lastname, A. B. & Coauthor, C. D. (2018). Title of paper. In Proceedings of the 14th International Conference on Precision Agriculture (unpaginated, online). Monticello, IL: International Society of Precision Agriculture.

Keywords. *Digital Agriculture, Machinery, Labor, Data, Real-time, Open source, Tracking,*

Introduction and Motivation

The realm of digital agriculture, the fusion of digital communication and computational technologies with physical phenomenon to build data sets and a knowledge base for improved decision making, offers tremendous potential for improving both short-term and long-term performance in agricultural systems. While the long-term benefits may bring about the dramatic shifts in performance via reduced inputs, more efficient use of labor, energy, and machinery and even environmental benefits, a “promise” is not enough motivation for real people to expend precious time and energy to construct robust data sets. As we increasingly move toward machine learning and artificial intelligence to analyze the large data sets that we can generate, those data sets will need much more contextual information and quality characterization. That type of information is often too tedious to generate – and with the lack of immediate payback, the quality of data will suffer.

For this reason, we hypothesize that better data quality all but requires that the data generated have value in real or near-real time. If the data has value to workers and managers in the moment, there will be direct motivation to improve it. Furthermore, while it is a great intention of many to “write things down” later, as the flow of a workday goes, those moments (and the required memory of details we knew earlier) just seem to vanish.

To generate applications and architectures for data acquisition, storage, and visualization in a proprietary manner simply requires too much original coding. The open source movement which brings nearly all of our web-enabled mobile applications has an infrastructure ideally suited to help advance digital agriculture more quickly. By using these resources to find suitable libraries, import time-tested and widely accepted code, and more rapidly deploy applications, we can focus our attention more on the problems to be solved rather than building a code-base from scratch. More details about specifics in this area are in the sections below as they relate to specific applications.

Data in agricultural applications can have a wide range in resolution and precision, come in a wide variety of formats, and move with assorted interfaces. In this work we purposefully strive to further develop and demonstrate the OADA API (Open Ag Data Alliance, 2018) as well as other APIs which use common storage platforms with which many in and out of agriculture are familiar. When the data rate is low, these other APIs can work extremely well. As data rate increases, there may be a need for microservices and websockets to stream data as found in the OADA API.

At the core of the applications presented herein is one of the definitions of “transaction”: *a communicative activity involving two things that reciprocally affect each other.* It is a transaction when a machine alters soil, when a person treats or measures a steer, when a machine uses a machine to collect an object, a machine distributes materials, etc. These transactions, however small, can be documented easily and seamlessly with mobile communication and computing technologies. This type of transactional data has huge potential for improving logistics, management, and decisions. Consistent with the vision laid out by Steinberger et al. (2009), farm equipment can be an important source of knowledge for data-based plant production.

Approach:

The work presented below is entirely and freely available in github repositories as open source projects. The goal of these individual applications was to generate real-time data and at least some visualization and analysis bringing immediate value with minimal human involvement. In several of these cases, the data collected is not available on a low-cost sensor; for these situations, a simple user interface is used to keep the interruption to “regular work” as low as possible. The real-time analysis and visualization in these examples is a combination of local processing and cloud computing.

The applications described below use many public libraries. Some of the more valuable elements include:

- Numeral.js for formatting numbers
- Moment.js for formatting dates
- React.js for tying user interfaces to state changes
- Cerebral.js for application state management
- Google API JavaScript library for interacting with Google maps and Google sheets
- Trello JavaScript library for interacting with Trello boards/lists/cards
- Create-react-app to setup project templates easily
- Webpack for handling the bundling and development of JavaScript applications
- Babel to compile modern JavaScript in a backward compatible manner for browsers
- Github pages for hosting the apps
- Leaflet.js for mobile friendly maps
- Csvjson for data translation
- Pouchdb for database synchronization and caching
- Material-ui for clean styling and layout
- oada-srvc-docker to run a microservice style API

While different data repositories and file types pose challenges for interoperability, undoubtedly these challenges will persist. The open source approach should alleviate much of the hassle since others can readily acquire the code, API specs, and data formats (Brewster et al., 2017).

Sample Applications:

Several applications are explained below with screen captures to illustrate features and functionality. They are summarized in Table 1.

Livestock Treatments

Collection of the livestock treatments data must be done in real time. There is no reasonable manner to get this information after-the-fact. Once the data is collected, analysis of effectiveness of treatments, inventory tracking of supplies, and “history of each animal” is readily possible. The livestock treatments app has an interface which minimized taps and keystrokes to get the transactional data recorded efficiently (Ault, 2018c; Ault, 2018d). As shown in Figures 1 and 2, the interface is button based. The abbreviations are seemingly cryptic, but understandable and logical to the user intimate with the treatments indicated. Using typical mobile device operating practices, recent history is retained as generally there are common input sets that a user will repeat. A good user interface will maximize use of color and include buttons appropriately sized for mobile devices.

Table 1. Open source* applications which demonstrate methods and value of real-time data collection, visualization, and exchange.

Application Name	Purpose	Format	Data Storage
Livestock Treatments	Accurately and quickly record livestock treatments and assess antibiotic sensitivity per animal	Web app	Trello
Livestock Weight	Document cattle weight to facilitate real-time sorting based on rate of gain in a production pipeline	Web app	Google Sheets and Trello
Rock	Note location of rocks to collect among the field work team	Web app or Android	Local with cloud sync
Fieldwork	Serves as planner, logistics facilitator, and record keeper for field operations	Web app or Android	Local with cloud sync
Manure	Records of application rate, date, location, source, operator	Android app	Local with cloud database sync
TrialsTracker	Note taking during crop production season to facilitate comparison and analysis of treatments (planned and unplanned)	Web app	Local with cloud database sync

* https://github.com/OpenATK

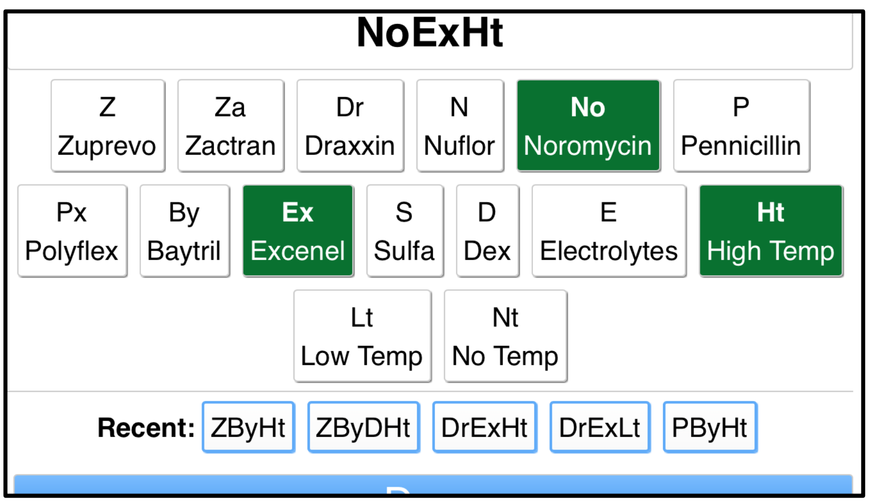

Fig 1. Livestock treatment app treatment selector view.

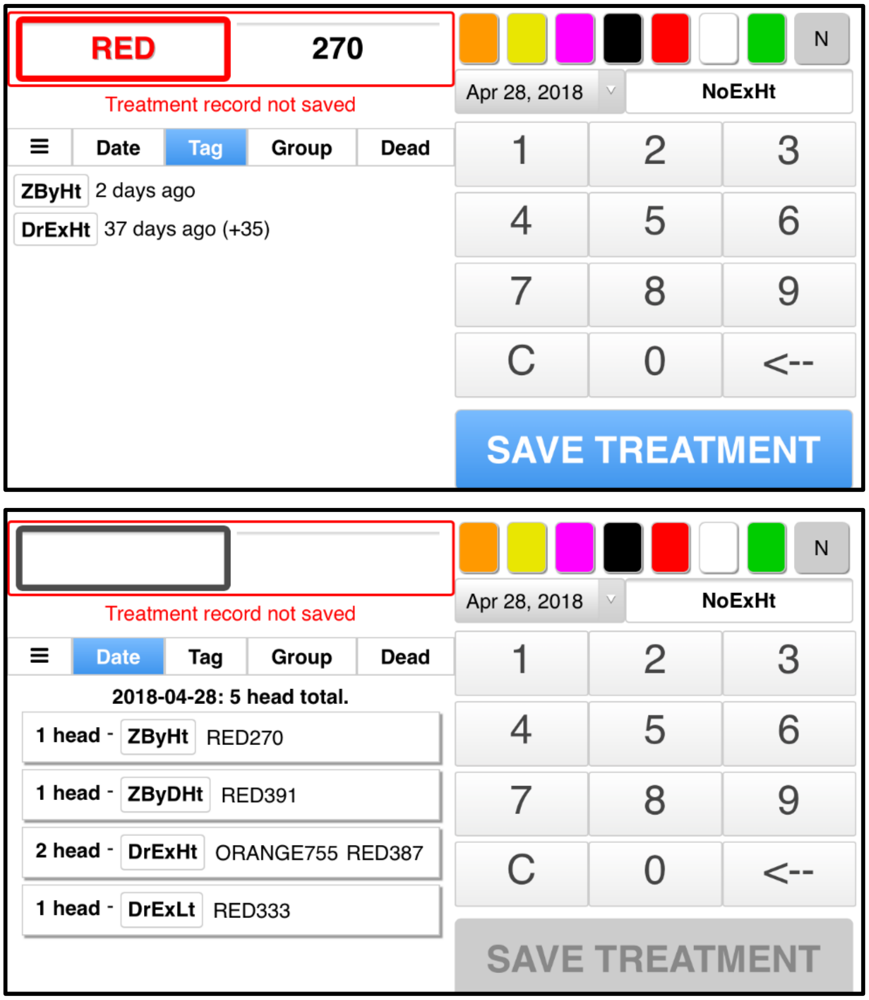

Fig 2. Livestock treatment app, tag entry view (top) and date summary view (bottom).

The back-end storage of data for the livestock treatment app is Trello (Trello, 2018; Figure 3). Trello is a private (yet shareable) cloud storage environment which is structured with a 4-tier organization hierarchy beginning with boards; boards have lists; lists have cards. Cards can contain text, comments, photos, files, and hyperlinks. Changes to Trello items are tracked with date/time stamps. As a result, the complete transaction history can be told by examining the metadata associated with cards. Consistent formatting of cards (which is automatic with data coming via the app) can facilitate searches and data mining of the information contained. With Business Class, Trello data can be exported to CSV (comma separated values) or JSON (JavaScript Object Notation) formats, or accessed as JSON via the Trello API on the free tier.

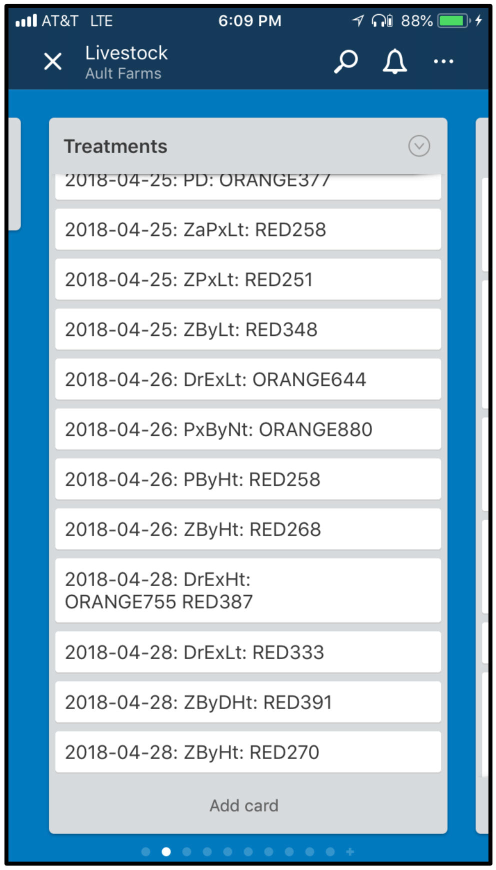

Fig 3. Livestock treatment app, Trello backend storage showing one card per animal tag/date combination.

Ault (2017) briefly explained the livestock treatments app and the advantage of the open source development. With 1400 lines of code written in 20 hours added to nearly 300,000 lines of other code from 764 libraries, the app was born.

Livestock weighing

When sorting cattle for sale, the common metrics are simply body condition (visual analysis) and body weight. However, with rate of gain information, a stock manager can make better decisions to keep animals continuing to gain at an acceptable rate. However, when sorting cattle at a production facility with thousands of head, there are few precious seconds to make decisions regarding each animal. With previous weight/date information and real-time weight/date data, simple calculations can provide this insight. The livestock weighing app enables this insight (Ault, 2018a; Ault, 2018b).

As illustrated in Figure 4, the livestock weighing app is a two-cursor design to support a pipeline of single-file cattle awaiting entry to the scale; tags can be entered in order independent of weights. There is an entry for the tag number and the current weight (which would come from a calibrated scale). Using the Google API (Google, 2018), data is imported and retrieved from Google Sheets (Figure 5) to display rate of gain in real time.

As one example of the importance of real-time beyond the sorting itself is that this data is displayed in real-time. An unreasonable rate of gain would be observed immediately. If that happened to be caused by user entry error, it could be corrected immediately. If that happened to be caused by scale error, it too could be corrected and the collection of an entire set of bad data would be avoided. In fact, this occurred on just the second use of the app in production when it was discovered that one group of calves was given the wrong set of ear tags after they failed to show up with appropriate rates of gain in the app. Without real-time analysis and display, poor data could proliferate.

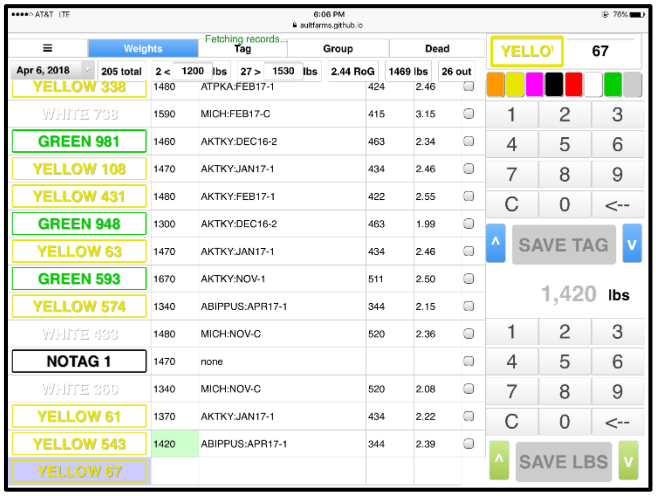

Fig 4. Livestock weighing app, double-cursor design with data entry view showing rate of gain in real time to aid in sorting for sale.

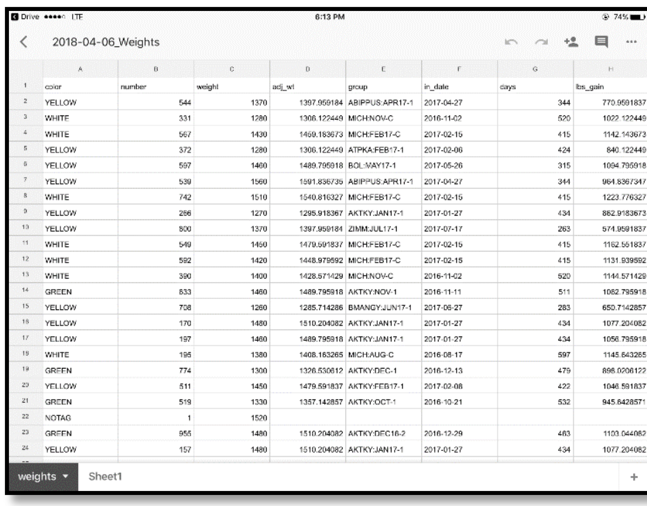

Fig 5. Livestock weighing app, Google sheets backend storage showing summary of date, weight, gain by tag number.

Rock app

One of the common activities on many farms is the identification and later collection of rocks which can damage soil engaging equipment. In most instances, a rock that needs to be removed would be identified and located at an inopportune time for collection. It is unreasonable to expect farm workers to remember details of these locations and other relevant details (such as size, need for extraction). With a very simple map-based and graphical interface, the rock app is a tool for noting this information in real-time in a manner shared among farm employees (Fiechter, 2018; Open Ag Toolkit, 2014; OpenATK, 2018c; Welte, 2013). Figure 6 illustrates the web app interface which includes ability to declare (and edit) a location and add a comment. To note that a rock has been collected (picked up), a user simply taps the specific rock icon and chooses “pick up”. The icon changes color to indicate the activity.

The Rock app is also available on GooglePlay for Android devices (Open Ag Toolkit, 2014). This version uses Trello as the backend storage with a Trello sync app needed (OpenATK, 2018b).

The Rock app is a specific instance of an item marking tool which could be generalized to many agricultural (and other) applications. For instance, location and status of liquid petroleum (LP) tanks, anhydrous ammonia (NH_3) tanks or trailers, loaned equipment, drainage tile needing fixed, trees/brush to be removed/trimmed, etc. could be noted and shared among trusted employees.

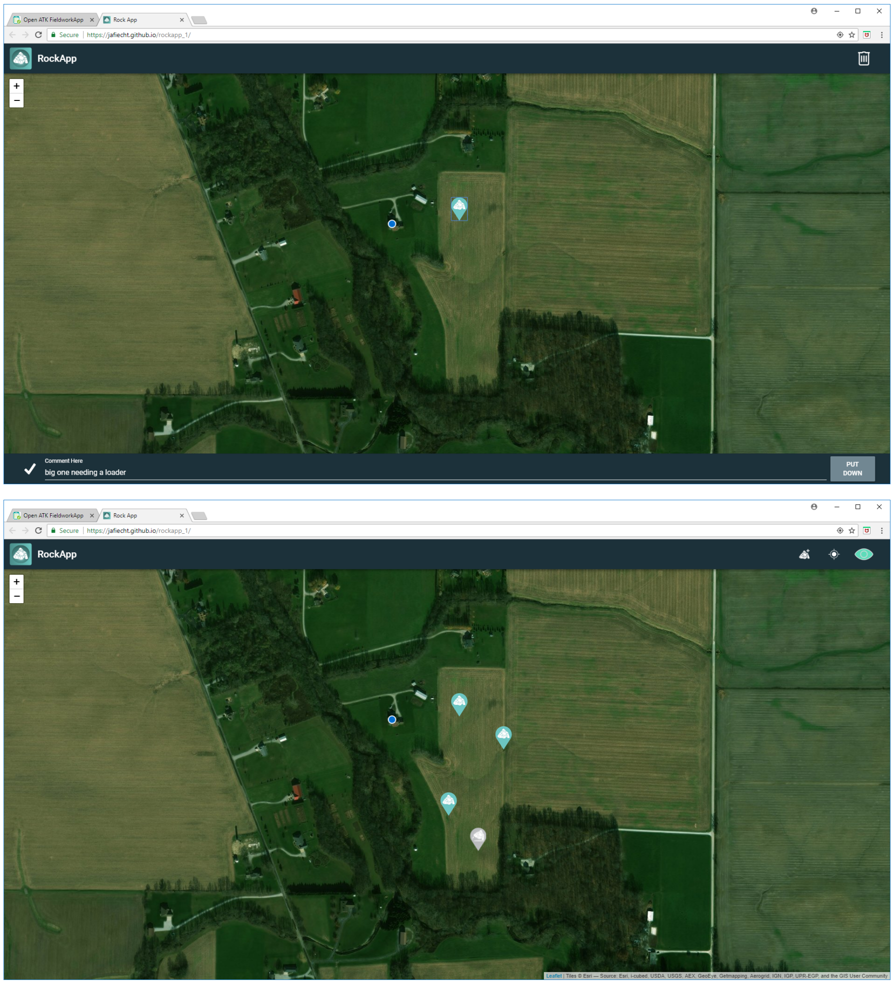

Fig 6. Rock app, view (top) showing comment block for the selected rock and view (bottom) showing several rocks in the field with one of four already picked up.

Fieldwork app

There are many software packages available for precision agriculture, but these geo-spatially referenced tools do not easily assist with regard to planning field activities or tracking progress through the season. The Fieldwork app offers a simple interface to share field boundaries, do field-by-field planning of multiple activities through the year, and track the progress in each activity (such as % planting completed on May 14; Open Ag Toolkit, 2015; OpenATK, 2018a; OpenATK, 2018c). Fieldwork via webapp is illustrated in Figure 7. The real-time nature of data is critical for managing logistics of machinery and workers throughout the busiest times of the farming cycle. With brief comments entered with the activity record, deeper mining of the data is possible later.

The Fieldwork app is also available for Android devices on GooglePlay (Open Ag Toolkit, 2015). It, too, uses the Trello sync app (OpenATK, 2018b) to align the interface with Trello cards. There are separate "settings" lists for each operator (name) and field (name, area, and boundary coordinates) which populate menu choices; there are separate cards for each field activity and these cards indicate the field, operator, activity, and whether the activity is planned, started, or completed.

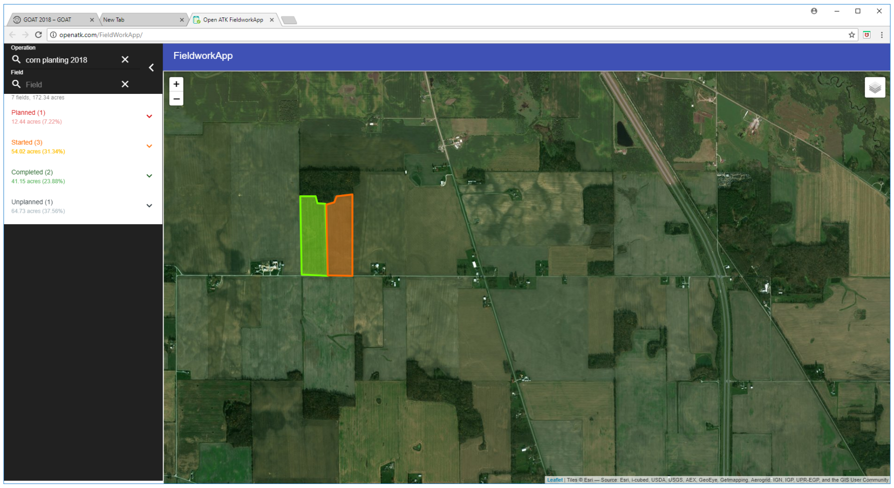

Fig 7. Fieldwork app used to track progress (planned, started, completed) of different field operations by different workers on defined fields.

Manure app

The manure app was developed as a demonstration of an app to generate data automatically – without user intervention (Koester, 2015; OpenATK 2015; OpenATK 2018c); it has a manual mode of operation which may have larger application and offers these features:

- give advice to the user as to the target speed to apply the correct rate
- record, by the load, the operator, location, time, source, and rate of manure application
- record in map view the path for each load applied
- optionally auto-detect spreader ID and status (on/off)
- optionally auto-detect field ID (by geo-fence of field boundaries)

To provide this information, there is a minimal amount of setup required (Figure 8); there is information needed regarding the spreader, field, source, and operator. In “manual” mode, the operator simply taps “Start Unloading” when the spreader is spreading. The app will change view and then offer an icon to “Stop Unloading”. In the meantime, the path and time are recorded and a new record is generated. Figure 9 illustrates the abbreviated view of the record. The map view is not shown but will display the path of each load. This is particularly helpful if spreading is done by different operators or in different time spans since it will show exactly where spreading already occurred.

To enable the fully automatic mode, a Bluetooth tag with accelerometers is required. The tag identifies which spreader is connected to the tractor and through the use of the accelerometers, the app will know if the spreader is on (spreading) or off (not spreading). The fully automatic mode will also use the geo-fence of the field boundaries to automatically identify the field. The field of the current location and the current spreader will influence the target speed since application rate is a function of target nutrient application, source characteristics, and spreading width.

The by-the-load records of manure spreading can be exported to csv for analysis or summary in spreadsheet software. This will, no-doubt, be helpful in generating future required reports regarding nutrient management and actual implementation.

While the manure app is specific to records involving solid or liquid livestock waste, the principles therein are applicable to all field operations of distribution (spreading, spraying, or any application of materials). Similarly, any gathering/harvest operations which are in a batch mode (such as baling, module building, or hauling by loads) could be tracked with a very similar application.

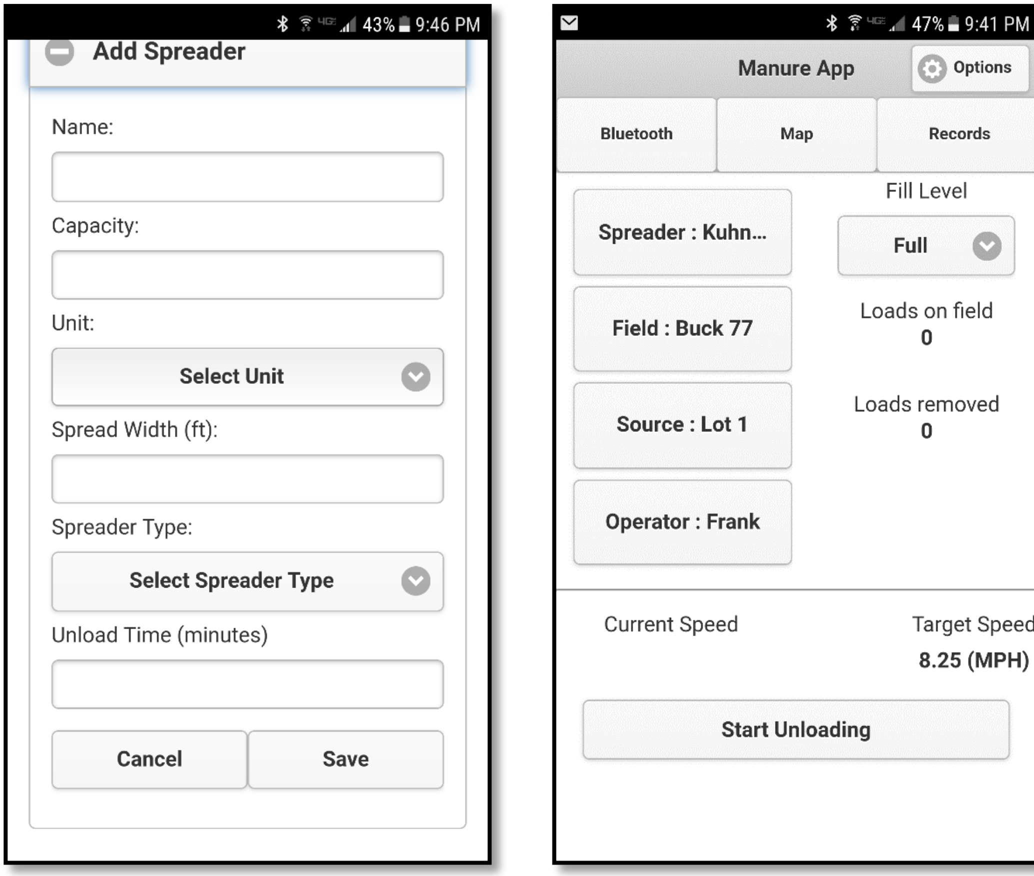

Fig 8. Manure app screens (left) used to add a spreader and describe spreader characteristics and (right) offering immediate operation speed suggestion and ready to track application.

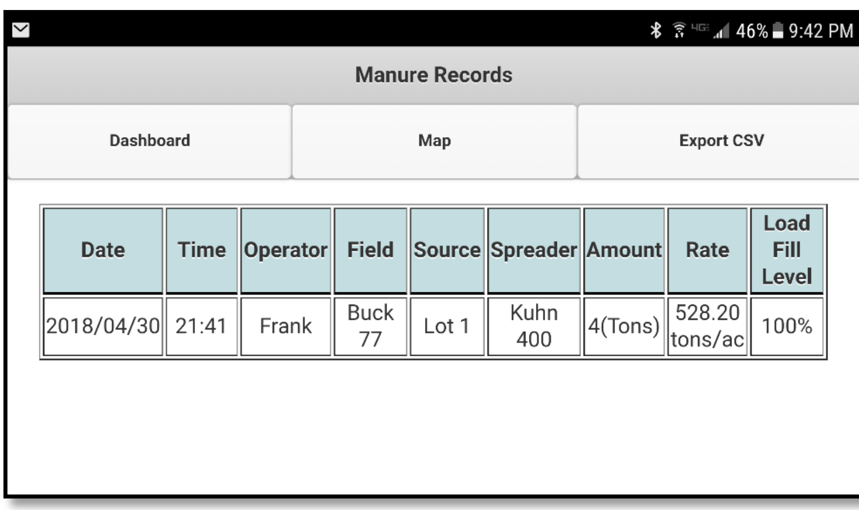

Fig 9. Manure app output report summary sample (also available via csv for analysis)

TrialsTracker app

Increasingly there is a need and desire to perform on-farm, in-field experiments to compare management strategies and crop or soil treatments. Some of these experiments are planned and some happen as a byproduct of other events (like running out of a certain hybrid, delay due to weather, switching anhydrous tanks to one with or without N-Serve). For these experiments, it is important to note what/when/how/why/where. The TrialsTracker app was developed for this purpose and offers a minimal-tap interface to document areas, tag/flag treatments, and analyze impacts (Figure 10; Noel, 2018; OpenATK, 2018c). The original intent was to analyze yield data (bu/ac or kg/ha), but the same framework can be used to analyze other layers of data such as fuel consumption, wheel slip, or implement draft. This app leverages data made available via an instance of the OADA API (Open Ag Data Alliance, 2018).

The TrialsTracker app uses data stored in geohashed buckets for rapid display according to the zoom level of the view. The aggregate by finger as illustrated brings rapid comparisons of valuable metrics to light quickly (Figure 10).

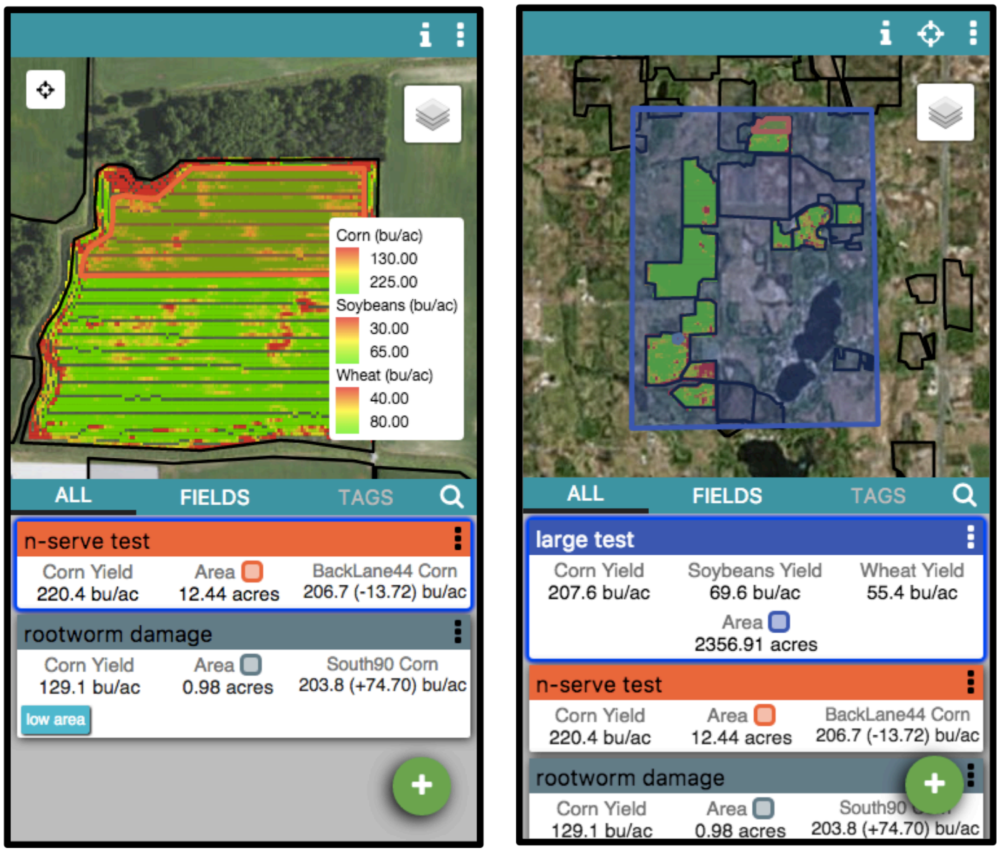

Fig 10. Trials Tracker app showing (left) comparison of a test area to the balance of the field and (right) summary view of selected region.

Summary:

In agricultural work settings, timing is generally very important and seamless collection of data is critical if that data is to be collected. In the absence of IoT sensors, this requires some human involvement; in these cases, the user interfaces must be efficient. Perhaps more importantly, the real-time nature of data utilization can influence data capture (if at all) and data quality. If data is useful in the near term (including almost immediately), then that data will most certainly be collected and can be checked for accuracy. Examples of data collection in livestock and cropping systems were presented with varying back-end storage options and with varied levels of complexity. All applications presented are open source and readily adaptable to similar situations with applicable specificity and focus.

Acknowledgements:

This work was supported by the USDA National Institute of Food and Agriculture as well as Purdue University, Winfield, The Climate Corporation, and CNH Industrial.

References:

Ault, A. 2017. *2017 11 17 OATSPrelaunch AultOpenSource.* https://www.youtube.com/watch?v=Ii21Hq_j7gI

Ault, A. 2018a. Af-monorepo. https://github.com/aultfarms/af-monorepo. Accessed 5/1/18.

Ault, A. 2018b. Ault Farms livestock weighing web application. https://aultfarms.github.io/af-monorepo. Accessed 5/1/18

Ault, A. 2018c. Ault Farms – Treatments. https://aultac.github.io/treatments/#. Accessed 5/1/18.

Ault, A. 2018d. Treatments. https://github.com/aultac/treatments. Accessed 5/1/18.

Brewster, C., I. Roussaki, N. Kalatzis, K. Doolin, and K. Ellis. 2017. IoT in Agriculture: Designing a Europe-Wide Large-Scale Pilot. *IEEE Communications Magazine.* Sept. 26-33. doi: 10.1109/MCOM.2017.1600528

Fiechter, J. 2018. RockApp. https://jafiecht.github.io/rockapp_1/. Accessed 5/1/18.

Google, 2018. https://developers.google.com/sheets/api/

Koester, M.R. 2015. *A nearly autonomous, platform-independent mobile app for manure application records*. M.S. Thesis in Agricultural and Biological Engineering. Purdue University. West Lafayette, IN. 107 pp. https://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1501&context=open_access_theses. Accessed 5/1/18.

Noel, S. 2018. Trials Tracker web interface. https://trialstracker.oada-dev.com/. Accessed 5/1/18.

Open Ag Data Alliance. 2018. Open Ag Data Alliance. https://github.com/oada. Accessed 5/1/18.

Open Ag Toolkit. 2014. Rock – OpenATK. https://play.google.com/store/apps/details?id=com.openatk.rockapp. Accessed 5/1/18.

Open Ag Toolkit. 2015. Field Work – OpenATK. https://play.google.com/store/apps/details?id=com.openatk.field_work. Accessed 5/1/18.

OpenATK. 2015. Manure App. https://github.com/OpenATK/ManureApp/blob/master/ManureApp.apk. Accessed 5/1/18.

OpenATK. 2018a. FieldworkApp. http://openatk.com/FieldWorkApp/. Accessed 5/1/18.

OpenATK. 2018b. OATK Trello Sync. https://github.com/OpenATK/Trello/blob/master/Trello_Sync_OpenATK_v2.3.apk. Accessed 5/1/18.

OpenATK. 2018c. Open Ag Toolkit. https://github.com/openatk. Accessed 5/1/18.

Steinberger, G. M. Rothmund, and H. Auernhammer. 2009. Mobile farm equipment as a data source in an agricultural service architecture. *Computers and Electronics in Agriculture*. 65:238-246.

Trello. 2008. https://developers.trello.com/reference/. Accessed 5/1/18.

Welte, J., A. Ault, C. Bowman, S. Ellis, D. Buckmaster, D. Ess, J. Krogmeier. 2013. An approach to farm management information systems using task-specific, collaborative mobile apps and cloud storage services. ASABE Paper 131579954. American Society of Agricultural and Biological Engineers. St. Joseph, MI. 24 pp.