FAIMS Mobile: Flexible, open-source software for field research

Brian Ballsun-Stanton[a], Shawn A. Ross[b,*], Adela Sobotkova[a], Penny Crook[c]

[a]*Department of Ancient History, Macquarie University, Sydney, Australia*
[b]*Big History Institute, Department of Modern History, Politics, and International Relations, and Department of Ancient History, Macquarie University, Sydney, Australia*
[c]*Department of Archaeology, La Trobe University, Melbourne, Australia*

Abstract

FAIMS Mobile is a native Android application supported by an Ubuntu server facilitating human-mediated field research across disciplines. It consists of ‘core’ Java and Ruby software providing a platform for data capture, which can be deeply customised using ‘definition packets’ consisting of XML documents (data schema and UI) and Beanshell scripts (automation). Definition packets can be autogenerated using an XML-based domain specific language. FAIMS Mobile includes features allowing rich and efficient data capture tailored to the needs of fieldwork. It also promotes synthetic research and improves transparency and reproducibility through the production of comprehensive datasets that can be mapped to vocabularies or ontologies as they are created.

Keywords: Android, Mobile software, Field research, Archaeology, Geoscience, Ecology, Oral history

1. Motivation and significance

Many disciplines in the social sciences, humanities, and biological, earth, and environmental sciences depend upon data generated through human-mediated fieldwork. Such data might arise from excavation in archaeology, wildlife observation in ecology, soil sampling in environmental geochemistry,

*Corresponding author. Department of Ancient History, W6A 510, Macquarie University 2109 NSW, Australia. Tel.: +61 9850 7085.

Email addresses: `brian.ballsun-stanton@mq.edu.au` (Brian Ballsun-Stanton), `shawn.ross@mq.edu.au` (Shawn A. Ross), `adela.sobotkova@mq.edu.au` (Adela Sobotkova), `p.crook@latrobe.edu.au` (Penny Crook)

URL: `http://www.faims.edu.au` (Brian Ballsun-Stanton)

or subject interviews in oral history. Field research disciplines, however, often lack transparency and reproducibility, compromising the integrity of research results [1]. Field data is often collected using an ad-hoc mix of hard copy, data fragments in various formats, and bespoke databases [2, 3, 4, 5]. Datasets, furthermore, are often trapped in hard-copy archives, local storage, or digital 'silos', making them difficult to discover and limiting reinterpretation and reuse [6]. Digital datasets are often highly variable, of poor quality, and incompatible. Deficiencies like these inhibit re-analyses of primary data and the combination of datasets from multiple studies for large-scale research [4, 7, 1].

Insufficient attention has been paid to the development of software specifically designed for digital data collection during field research. Some tools exist for discrete tasks, such as measuring strikes and dips for structural geology (e.g., GeoCline or Rocklogger for Android), but more complex and flexible field data collection has been neglected. Most digital data collection in archaeology, for example, is accomplished either using a combination of generic and repurposed mobile and desktop applications (e.g., multimedia, office productivity, GIS, database, or questionnaire / survey software), or by building bespoke applications. Both approaches have severe limitations [8]. Bespoke software is expensive to build and maintain, placing it beyond the reach of all but the best-funded projects and organisations (e.g., iDig, created by the American School of Classical Studies at Athens: `http://idig.tips/` (Archived at: `https://perma.cc/23PS-6567`); [9]). Repurposed software requires field researchers to make do with applications, designed for other contexts, which lack critical features but still require extensive customisation (cf. the use of a suite of iOS applications at Pompeii [10], or Ben Carter's combination of Kobo Toolbox, PostGIS, QGIS, LibreOffice Base, and pgadminIII [11]).

FAIMS Mobile, conversely, is 'generalised' software which combines the particular features required for field research with sufficient customisability and redeployability to allow its use across disciplines, providing a large enough user base to support its development and maintenance and have a meaningful impact on research (see Section 4 below; cf. [8]). FAIMS Mobile is open source software developed by the Field Acquired Information Systems Project, an e-research infrastructure project based at Macquarie University, Sydney, Australia. It is mature software that has been under development since 2012 (see: [12, 13, 14]). Most other generalised field data collection software used for fieldwork, such as ARK, Heurist, or Kora [15], requires a continuous connection to a server.

FAIMS is most comparable to Open Data Kit (ODK) and its variants, but is differentiated by its lineage. ODK, another mature offline mobile data collection platform, was designed for social surveys, where an investigator asks questions of a interviewee. FAIMS, conversely, originated in archaeology, where an investigator records observations about things in the material world, relationships between those observations, and metadata contextualising the collection of those observations. Both projects are open-source, Java-based data collection platforms with similar potential and shared libraries, serving researchers who face similar problems; while solutions to those problems diverged significantly when FAIMS began development, over time features have converged to a degree. Features specific to FAIMS include mature bi-directional synchronisation across all devices (an alpha feature in ODK 2.0), use of an append-only datastore that provides a version history for all records, support for a wider range of external sensors, and more advanced GIS data operations (compared to GeoODK). Provisions for help and metadata capture are also richer and more granular. Field research projects, especially in liminal disciplines such as linguistics or oral history, would be wise to evaluate both platforms.

1.1. Experimental setting

FAIMS Mobile is designed to collect heterogenous data of various types (structured, free text, geospatial, multimedia) produced by arbitrary methodologies during human-mediated field research. It requires customisation to instantiate a project-specific data model, user interface, and workflow, but it addresses problems shared across field-based projects, such as provision of a mobile GIS and automated synchronisation across multiple devices in a network-degraded environment.

During a typical deployment, researchers work with FAIMS project staff to articulate their data model and workflow. A FAIMS developer then renders that methodology into a ‘definition packet’ of files that produce a ‘module’ (i.e., an implementation of FAIMS Mobile customised for a particular project). Separate definition packet files control the data schema (XML), the user interface (XML and CSS), and automation and logic (Beanshell), offering nuanced control. The interface can also be translated into multiple languages using a (plain text) localisation file. Completed modules are then deployed to a local or online Ubuntu server, and from there onto as many Android devices as needed (after the core mobile application is installed, e.g. from Google Play). Data is then collected using those devices, which can operate fully offline, and synchronised opportunistically when a network connection to the server is available. Data can be validated at the time of entry on the device, or later on the server. At the end of data collection, data is exported in the user’s desired format by means of a customisable exporter. Three deployment case studies have been published in Sobotkova, et al., 2016[8].

Alternatively, FAIMS has developed a XML-based domain specific language (DSL) to simplify customisation. Using this DSL, a single file can be used to generate all necessary definition packet files. In addition to deployments conducted by the FAIMS team, projects have independently customised FAIMS Mobile themselves using both approaches [16, 17].

2. Software description

FAIMS Mobile is open-source, customisable software designed specifically to support field research across many domains. It allows offline collection of structured, text, multimedia, and geospatial data on multiple Android devices, and is built around an append-only datastore that provides complete version histories. It includes customisable export to existing databases or in standard formats, supported by features that facilitate data compatibility. Finally, it is designed for rapid prototyping using and easy redeployability to reduce the costs of implementation.

2.1. Software Architecture

FAIMS Mobile consists of 'core' software written in Java and Ruby, customised to particular field deployments using reusable and sharable definition packets consisting of XML, Beanshell, and CSS files (or, by sacrificing some nuances of control, a single file written in a DSL). More specifically, FAIMS uses the following technologies:

- Javarosa to render native Android UI elements at runtime;
- Sqlite3 to store an attribute-key-value datastore (with data schemas definable at runtime);
- An append-only data model inspired by Google's Protobufs;
- Beanshell to provide runtime scripting via calls to an underlying Java API;
- Spatialite to encode geospatial data in the datastore;
- Nutiteq to render geospatial data;
- NativeCSS to style android-native elements;
- Antlr3 as a grammar parser for identifiers; and a

- Ruby on rails/Apache stack to provide a server, which can be hosted online or on modest hardware in the field.

We developed this architecture to meet two fundamental requirements: (1) the software had to accommodate a wide range of research designs, data schemas, and workflows, and (2) the software had to accommodate extremely variable structured, free text, multimedia, and geospatial data. Essentially, we needed to build a system capable of rendering and recording arbitrary field data, since individual 'data loggers' tied to a particular methodologies (even if extensible) would not be worth the investment to build and deploy as separate mobile applications.

Our Android client can, at runtime, render an arbitrary data collection methodology (schema and workflow), save all records to a datastore, and opportunistically synchronize that data with instances of the software running on other devices. This distinction is much like the one between a web browser and a website. A browser contains many sophisticated engines for rendering the page, its interactivity, and its styling, but does not have content. A website uses the HTML engine provided by the browser to display its specific content. FAIMS Mobile likewise provides engines for rendering definition packets to produce customised data collection modules.

Four years of deployment experience revealed the importance of quality assurance, something too often neglected in academic software [18, 19]. Each customisation and deployment is, indeed, a miniature software development project [8]. Due to the need for significant QA per deployment, FAIMS Mobile 2.5 supports Robotium for unit and integration tests on customised data collection modules, such that large amounts of test data can be automatically added via the normal user interface. This allows users to load test their modules under simulated field conditions.

2.2. Software Functionalities

FAIMS Mobile improves field research by providing a wide range of features that specifically address the needs of field research across disciplines, while facilitating the production of compatible datasets from heterogeneous data structures and workflows. These features include:

- Deep customisation of data schema, user interface, and automation using either a packet of XML, Beanshell, and CSS documents for nuanced control, or a single file in an XML-based domain-specific language for ease of deployment. Definition document(s) are separate for core software, making modification and reuse easier.

- Collection of various data types within a single record, including structured data, geospatial data, free text, sensor-produced multimedia, and file attachments.

- Automated, configurable synchronisation across an unlimited number of devices using a local or online server.

- Synchronisation is opportunistic, whenever a connection is available, allowing devices to work in network-degraded environments or offline for extended periods of time. Robust offline capability is achieved through maintenance of the datastore on each device, not caching.

- Defaults, flow logic, hierarchical selections, dynamic UI (expand, collapse, hide, or show input fields), and other advanced data collection features.

- Mobile GIS supporting raster and vector data, layer management, legacy data visualisation, and point, line, and polygon creation and editing. Multiple records can be linked to a single shape, or multiple shapes to a single record.

- 'Annotation' and 'certainty' fields attached to every record. The former allows the collection of granular metadata (mimicking the 'margins of the page' in paper recording), while the latter allows users to record their confidence in an observation.

- Internal and external sensor support, external Bluetooth devices like GPS receivers and USB / HID devices like digital balances and calipers.

- Multilingual support using a localisation file.

- An append-only datastore providing a full revision history, including the ability to review and reverse changes selectively.

- Mobile device and server-side validation.

- Aids to good practice including contextual HTML help, 'picture dictionaries' (selections based on images), and selection trees that can guide users through complex processes.

- Embedding of URIs into controlled vocabularies or other elements to link them to shared vocabularies, thesauri, or ontologies.

- Customisable export to desktop software, pre-existing databases, or online data services (based on SQL queries).

2.3. Sample code snippets analysis

While a thorough discussion of the module code is out of scope for this paper, we have two fundamental documents which discuss module creation from start to finish. The first: ‘FAIMS User to Developer Documentation’ (linked at `https://www.fedarch.org/support/#2`, archived at: `https://perma.cc/8JDY-6RKL`) is designed to walk normal users through the creation of a module from first principles. The second ‘The FAIMS Cookbook’ is a description of our data structures and API designed in a rough tutorial format (linked: from support page above, archived: `https://perma.cc/H6XJ-X6E2`).

3. Illustrative Examples

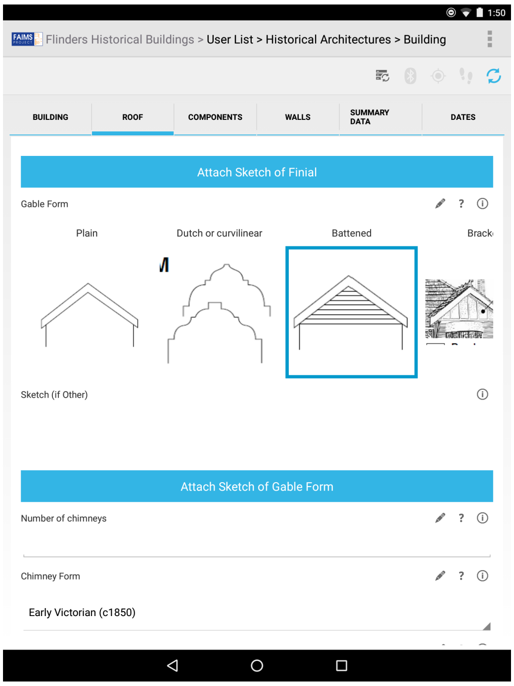

Figure 1: Structured data recording: including dropdowns, numeric fields, checkboxes, radio buttons, and ‘picture dictionaries’

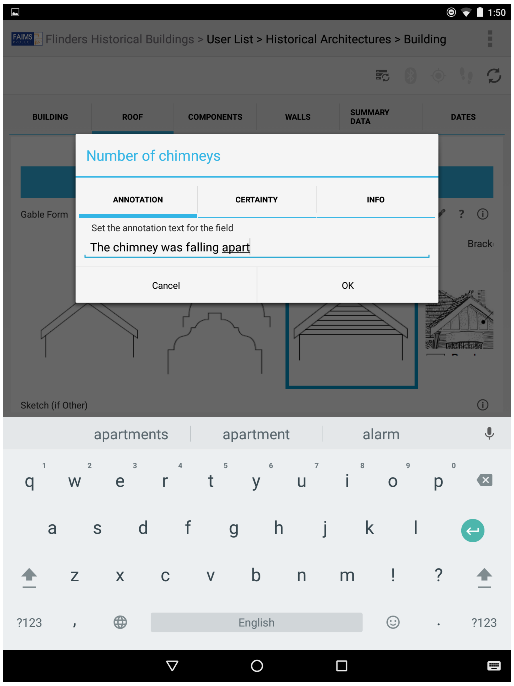

Figure 2: FAIMS has metadata such as annotations (digital ‘scribbling on the margin’) and certainty. Granular, contextualised, HTML-format help (‘info’) is also delivered using this interface

FAIMS offers a variety of ways to record data (Fig. 1) all of which can be arranged hierarchically. Each of the fields, regardless of datatype, also allows for the recording of metadata (Fig. 2).

Figure 3: FAIMS Mobile can render layers of georeferenced raster files like satellite images

Figure 4: And topo maps

Field research often requires spatial data capture and visualisation. FAIMS has GIS rendering capabilities for rasters (Figs. 3, 4), or vector data (blue polygons in both). Vector data can be created in the field and automatically bound to a record. Currently, most survey modules use mobile GIS functionality.

4. Impact

FAIMS allows the efficient collection of field data, dramatically reducing or eliminating manual digitisation (see [8]). Near-real-time availability of data from multiple devices for review also provides immediate error detection (especially when combined with validation and contextual help). Finally, the software is customisable, extensible, and community driven; if current or potential users request new features, they can be implemented (within budget constraints). Field research will never represent more than a fraction of the market for generic, mass-market products, whereas it is the sole focus of

FAIMS. Researchers can, therefore, compromise less and actively contribute to the development of purpose-built software through their specific customisations or core-software feature requests [8]. Organisations with sufficient development capacity are, of course, also welcome to contribute to the core software open source project.

Beyond the immediate needs of users, FAIMS Mobile improves research practice and data management. URIs can be embedded in controlled vocabularies and other elements[14], linking them to linked open data sources (e.g., species information can be linked to the Encyclopedia of Life[20]). Localisation can be used to 'translate' a local language of practice to a standard vocabulary (e.g., 'context' or 'locus' can be translated to 'statigraphic unit' - and then linked to an online ontology). Generalised export matches output to existing services or standards (e.g., archaeological records can be exported not only as shapefiles, CSVs, or a 3NF relational database for incorporation into an existing geodatabase, but also as XML or GeoJSON for ingest into domain-specific repositories like Open Context). Perhaps most importantly, comprehensive, rather than selective, datasets can be created and exported for publication, improving transparency and reproducibility. Combined with features that improve data compatibility across projects, large-scale and synthetic research become more feasible.

FAIMS Mobile also makes digital recording a more feasible and less costly option for researchers [8, 14]. The core software does the 'heavy lifting' of field recording (data storage, bi-directional synchronisation, GIS, etc.), and can be customised either leveraging the control of a full definition packet or the efficiency of the single-file DSL module generator. As a result, an experienced developer can rapidly prototype a recording system so long as data and workflow models are available (well-scoped field recording systems of moderate complexity can be prototyped in one to two developer-days). Reuse and modification of existing customisations from a growing, openly-licensed online library (leveraging version control systems like GitHub) also help reduce deployment costs[12]. Deployment of FAIMS Mobile is therefore less expensive than production of bespoke mobile applications, and competitive with deployment of a suite of generic tools - which would likely include a geodatabase, GIS, social survey software, multimedia management software, note-taking software, etc. [11] - while offering more functionality specific to field research, better integrating different data types, and requiring fewer compromises on the part of the researcher. Since FAIMS is also easier to redeploy than customised combinations of mass-market software, is allows improvements and innovations to be shared more readily[12].

FAIMS Mobile has changed users' daily practice. Three case studies involving archaeological deployments [8] indicate that users benefit from the increased efficiency of fieldwork, in that the time saved by avoiding digitisation more than offsets the time required to implement FAIMS and more data collected during fieldwork of a given length. Born-digital data avoided problems with delayed digitisation, which often occurred long after field recording when the context of of the record had been forgotten (or the person making the record was no longer available). Researchers reported more complete, consistent, and granular data, while information could be exchanged more quickly between excavators and specialists, which in one case improved ‘post-excavation reconstruction of the site’ and facilitated the evaluation of patterns for meaning in another. They also observed that the process of moving from paper to digital required comprehensive reviews of field practice, during which knowledge implicit in existing systems to become explicit and data was modelled more carefully. By participating in a ‘miniature software development project’, researchers gained familiarity with the strengths, limits, and demands of software deployment, especially the need for extensive testing. The greatest challenge posed by the transition from paper has been the reallocation of time from the end of a project (digitisation) to the beginning (data modelling, development, and testing), even if they realise an overall time savings.

Although adoption of digital recording during fieldwork represents a significant socio-technical change, FAIMS Mobile has seen good uptake. Since 2012 FAIMS Mobile has supported over 25 major research projects, with approximately 300 users logging over 10,000 hours in the application. Most uptake to date has been at large, multi-year projects that are still early in their lifecycle, so all FAIMS-related publications to date have focused on the software itself or the transition from paper-based to digital workflows. While archaeologists comprise the main user group, FAIMS now supports research in other disciplines as well. Fifteen archaeology, ecology, and history projects are scheduled for 2017 with an estimated usage of 12,000 hours. A 2016-2017 New South Wales Research Attraction and Acceleration Program award is funding links to government resources (e.g., automated data submission to the Aboriginal Heritage Information Management System of NSW) making it more attractive to commercial users. This award is also funding community-based heritage and science deployments, where members of the public will be able to download preconfigured versions of FAIMS Mobile to report information about archaeological remains or wildlife.

5. Conclusions

When they collect data digitally, field researchers often re-purpose mass-market or general-purpose software that was not specifically designed to meet their needs, often requiring several pieces of software (some of which are individually complex) to accommodate the rich and varied data they must collect. FAIMS Mobile offers an alternative, purpose-built for field research with extensive community input, including five years of iterative co-development with field researchers first in archaeology, and more recently in geoscience, history, and ecology. FAIMS Mobile offers an unparalleled range of features to support fieldwork, including collection of structured, free-text, multimedia, and geospatial data, deep customisability, mobile GIS, use of internal and external sensors, offline capability with opportunistic synchronisation using either an online or local server, full record version histories, multilingual support, certainties and annotations attached to individual fields, and rich contextual help. It includes customisable export to existing databases or in standard formats, supported by features that facilitate data compatibility. It is designed for rapid prototyping and easy redeployability to reduce the costs of implementation, leveraging community software version control systems like GitHub. In short, FAIMS Mobile is community-driven, customisable, extensible software that can support the socio-technical transition from paper to digital in field research disciplines and facilitate the production of comprehensive, compatible datasets to improve synthetic research, transparency, and reproducibility.

Acknowledgements

The FAIMS Project was funded during 2012-2015 by the National eResearch Collaboration Tools and Resources (NeCTAR) eResearch Tools program (RT043 and V005), from 2014 to 2016 by the Australian Research Council (ARC) Linkage Equipment, Infrastructure, and Facilities (LIEF) programme (LE140100151), and in 2017 by a Research Attraction and Acceleration Programme (RAAP) award offered by the New South Wales Government. The UNSW Australia (2012-2014) and Macquarie University (2015-present) hosted the project, making significant cash and in-kind contributions. Other organisations currently providing cash and in-kind contributions include: Flinders University, La Trobe University, the University of Queensland, UNSW Australia, the Commonwealth Scientific and Industrial Research Organisation (CSIRO), The Alexandria Archive Initiative (Open Context), Digital Antiquity (tDAR), and the University of York (the Archaeology Data Service). For a complete list of project sponsors and participants, see `https://www.faims.edu.au/`.

References

[1] M. McNutt, K. Lehnert, B. Hanson, B. A. Nosek, A. M. Ellison, J. L. King, Liberating field science samples and data, Science 351 (6277) (2016) 1024–1026.

[2] C. L. Borgman, Big Data, Little Data, No Data: Scholarship in the Networked World, MIT Press, 2015.

[3] E. C. Kansa, A. Bissell, Web syndication approaches for sharing primary data in "small science" domains, Data Science Journal 9 (2010) 42–53.

[4] K. Kintigh, The promise and challenge of archaeological data integration, Am. Antiq. 71 (3) (2006) 567–578.

[5] D. R. Snow, M. Gahegan, C. L. Giles, K. G. Hirth, G. R. Milner, P. Mitra, J. Z. Wang, Others, Cybertools and archaeology, Science 311 (5763) (2006) 958.

[6] T. Blanke, M. Hedges, A data research infrastructure for the arts and humanities, in: S. C. Lin, E. Yen (Eds.), Managed Grids and Cloud Systems in the Asia-Pacific Research Community, Springer, Boston, MA, 2010, pp. 179–191.

[7] K. W. Kintigh, J. H. Altschul, M. C. Beaudry, R. D. Drennan, A. P. Kinzig, T. A. Kohler, W. F. Limp, H. D. G. Maschner, W. K. Michener, T. R. Pauketat, P. Peregrine, J. A. Sabloff, T. J. Wilkinson, H. T. Wright, M. A. Zeder, Grand challenges for archaeology, Proc. Natl. Acad. Sci. U. S. A. 111 (3) (2014) 879–880.

[8] A. Sobotkova, S. Ross, B. Ballsun-Stanton, A. Fairbairn, J. Thompson, P. VanValkenburgh, Measure twice, cut once: Cooperative deployment of a generalized, Archaeology-Specific field data collection system, in: E. W. Averett, J. M. Gordon, D. B. Counts (Eds.), Mobilizing the Past for a Digital Future; the Potential of Digital Archaeology, The Digital Press at University of North Dakota, The University of North Dakota, Grand Forks, 2016, pp. 334–370.

[9] S. B. Fee, Reflections on custom mobile app development for archaeological data collection, in: E. W. Averett, J. M. Gordon, D. B. Counts (Eds.), Mobilizing the Past: Recent Approaches to Archaeological Fieldwork in the Digital Age, The Digital Press at University of North Dakota, 2016, pp. 219–234.

[10] S. J. R. Ellis, Are we ready for new (digital) ways to record archaeological fieldwork? a case study from pompeii, in: E. W. Averett, J. M. Gordon, D. B. Counts (Eds.), Mobilizing the Past: Recent Approaches to Archaeological Fieldwork in the Digital Age, The Digital Press at University of North Dakota, 2016, pp. 49–74.

[11] B. Carter, Crafting work flow- kobo toolbox/ PostGIS/ QGIS/ LibreOffice base/ pgadminIII — institute on digital archaeology method & practice, `http://digitalarchaeology.msu.edu/crafting-work-flow-kobo-toolbox-postgis-qgis-libreoffice-base-pgadminiii/`, archived at `https://perma.cc/5JPH-XQN4`, Accessed: 2017-2-25 (2016).

[12] S. Ross, B. Ballsun-Stanton, A. Sobotkova, P. Crook, Building the bazaar: Enhancing archaeological field recording through an open source approach, in: B. Edwards, A. Wilson (Eds.), Open Source Archaeology: Ethics and Practice, Versita, Warsaw, Poland, 2015, pp. 111–129.

[13] S. Ross, A. Sobotkova, B. Ballsun-Stanton, P. Crook, Creating eresearch tools for archaeologists; the federated archaeological information management systems project, Australian Archaeology 77 (2013) 107–119.

[14] A. Sobotkova, B. Ballsun-Stanton, S. Ross, P. Crook, Arbitrary offline data capture on all of your androids: The FAIMS mobile platform, in: A. Traviglia (Ed.), Across Space and Time. Papers from the 41st Annual Conference of Computer Applications and Quantitative Methods in Archaeology (CAA), Amsterdam University Press, 2015, pp. 80–88.

[15] J. M. Gordon, E. W. Averett, D. B. Counts, Mobile computing in archaeology: Exploring and interpreting current practices, in: E. W. Averett, J. M. Gordon, D. B. Counts (Eds.), Mobilizing the Past: Recent Approaches to Archaeological Fieldwork in the Digital Age, The Digital Press at University of North Dakota, 2016, pp. 1–32.

[16] J. Good, KPAAM-CAM overview, in: KPAAM-CAM – Crossroads Workshop, 2016, archieved at `https://perma.cc/VND8-X2JH`, Accessed: 2017-2-23.

[17] G. Kiley, Student-developed software streamlines archaeological analysis, `https://news.brown.edu/articles/2016/09/tablet`, archieved at `https://perma.cc/8MMX-94K3`, Accessed: 2017-2-23 (2016).

[18] M. Might, The CRAPL: An academic-strength open source license, http://matt.might.net/articles/crapl/, archived at: https://perma.cc/EMD3-Q98Z Accessed: 2017-2-25 (2010).

[19] Q. Sun, The scientific software developer in academia, http://www.software.ac.uk/blog/2012-05-01-scientific-software-developer-academia, archived at https://perma.cc/GMY7-BLH4, Accessed: 2017-2-27 (2012).

[20] E. O. Wilson, The encyclopedia of life, Trends Ecol. Evol. 18 (2) (2003) 77–80.

Required Metadata

Nr.	Code metadata description	Please fill in this column
C1	Current code version	2.5
C2	Permanent link to code/repository used for this code version	**Core Application** https://github.com/FAIMS/faims-android **Server** https://github.com/FAIMS/faims-web **Definition Packets** https://github.com/FAIMS
C3	Legal Code License	GPLv3
C4	Code versioning system used	git
C5	Software code languages, tools, and services used	Java, Ruby, XML, SQLite, Spatialite, Javarosa, Antlr, Puppet, Apache, Imagemagick, God, Beanshell, gson, guice, Nutiteq (non-free), NativeCSS, Protobuf, Robotium
C6	Compilation requirements, operating environments & dependencies	Android Studio, Ubuntu 16.04, Nutiteq license (for non-watermarked GIS)
C7	If available Link to developer documentation/manual	**Module Cookbook** https://faimsproject.atlassian.net/wiki/display/FAIMS/FAIMS+Data%2C+UI+and+Logic+Cook-Book **Module Beanshell API** https://faimsproject.atlassian.net/wiki/display/FAIMS/Program+Logic+Support **Developer documentation home** https://faimsproject.atlassian.net/wiki/spaces/FAIMS/overview **'User to developer' documentation** https://www.fedarch.org/support/
C8	Support email for questions	support@fedarch.org

Table .1: Code metadata (mandatory)

Nr.	(Executable) software metadata description	Please fill in this column
S1	Current software version	2.5.20
S2	Permanent link to executables of this version	**FAIMS Mobile** `http://www.fedarch.org/apk/` **Google Play** `https://play.google.com/store/apps/details?id=au.edu.faims.mq.fieldresearch2&hl=en` **Server Installer (wget and pipe to bash)** `https://raw.githubusercontent.com/FAIMS/faims-web/master/installer/puppetInstall.sh`
S3	Legal Software License	GPLv3
S4	Computing platforms/Operating Systems	Android, Ubuntu
S5	Installation requirements & dependencies	Android 6+, Ubuntu 16.04
S6	If available, link to user manual - if formally published include a reference to the publication in the reference list	'Getting started' guide and user documentation: `https://faimsproject.atlassian.net/wiki/display/FAIMS/Getting+started+with+FAIMS+-+an+overview`
S7	Support email for questions	support@fedarch.org

Table .2: Software metadata (optional)