

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 also be generated using an XML-based domain specific language, making customisation easier. 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, Field science

1. Motivation and significance

Many disciplines in the social sciences, humanities, and biological, earth, and environmental sciences depend upon data collected through human-mediated fieldwork. Such data might arise from excavation in archaeology, wildlife observation in ecology, soil sampling in environmental geochemistry, or subject interviews in oral history. Field research disciplines, however, often

*Corresponding author. Department of Ancient History, 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)

7 lack transparency and reproducibility, compromising the integrity of research
8 results [1]. Field data is often collected using an ad-hoc mix of hard copy, data
9 fragments in various formats, and bespoke databases [2, 3, 4, 5]. Datasets,
10 furthermore, are often trapped in hard-copy archives, local storage, or dig-
11 ital ‘silos’, making them difficult to discover and limiting reinterpretation
12 and reuse [6]. Digital datasets are often highly variable, of poor quality, and
13 incompatible. Deficiencies like these inhibit reuse of primary data and the
14 aggregation of datasets from multiple studies for large-scale research [4, 7, 1].

15 Insufficient attention has been paid to the development of software specif-
16 ically designed for digital data collection during field research. Some tools
17 exist for discrete tasks, such as measuring strikes and dips for structural ge-
18 ology (e.g., GeoCline or Rocklogger for Android), but more complex field
19 data collection has been neglected. Most digital data collection in archaeol-
20 ogy, for example, is accomplished either using a combination of generic and
21 repurposed mobile and desktop applications (e.g., multimedia, office pro-
22 ductivity, GIS, database, or social survey software), or by building bespoke
23 applications. Both approaches have severe limitations [8]. Bespoke software
24 is expensive to build and maintain, placing it beyond the reach of all but
25 the best-funded projects and organisations (e.g., iDig, created by the Amer-
26 ican School of Classical Studies at Athens: <http://idig.tips/> (Archived
27 at: <https://perma.cc/23PS-6567>); [9]). Repurposed software requires field
28 researchers to make do with applications, designed for other contexts, which
29 lack critical features but still require extensive customisation (cf. the use of
30 a suite of iOS applications at Pompeii [10], or Ben Carter’s combination of
31 KoBoToolbox, PostGIS, QGIS, LibreOffice Base, and pgadminIII [11]).

32 FAIMS Mobile, conversely, is ‘generalised’ software which combines fea-
33 tures required for field research with sufficient customisability to allow its
34 use across disciplines, appealing to a large enough user base to support its
35 development and have a meaningful impact on research (see Section 4 below;
36 cf. [8]). It is designed for offline use, unlike most other generalised field data
37 collection software such as ARK, Heurist, or Kora [12], which all require a
38 continuous connection to a server. FAIMS Mobile is open-source software
39 developed by the Field Acquired Information Management Systems Project,
40 an e-research infrastructure project based at Macquarie University, Sydney,
41 Australia. It is mature software that has been under development since 2012
42 (see: [13, 14, 15]).

43 FAIMS Mobile is most comparable to Open Data Kit (ODK) (<https://opendatakit.org/>, <https://perma.cc/9BGB-8RUT>) and its variants, but
44 is differentiated by its lineage. ODK was designed for social surveys, where an
45 investigator asks questions of a interviewee. FAIMS originated in archaeol-
46 ogy, where an investigator records observations about: things in the material
47

world, relationships between those observations, and metadata contextualising the collection of those observations. Both projects are open-source, Java-based data collection platforms customised using XML-based domain specific languages. ODK also offers simpler but more restrictive customisation using ODK Build (an HTML5 drag-and-drop interface), XLSForm (a tool that uses an Excel file to build a form), or third-party, GUI-based applications like KoBoToolbox. FAIMS, conversely, supports more profound customisation without modification of core software. It also includes features not found in ODK: more nuanced relationships between entities, bi-directional synchronisation across all devices (a feature in ODK 2.0 Tool Suite, which is in alpha release), use of an append-only datastore that provides a version history for all records, support for a wider range of external sensors and peripherals like label printers (‘ODK Sensors’ is in alpha release), a more sophisticated data export framework, and more advanced geospatial data operations (compared to GeoODK and its derivatives). FAIMS also has richer and more granular help and metadata capture. In short, FAIMS is more customisable and has more fieldwork-specific features than ODK, but as a result customisation is more entailed. 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), and accompanying metadata, produced using 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. The FAIMS Project provides customisation services under a typical open-source revenue model [16]. We also provide ‘User to Developer’ documentation (<https://github.com/FAIMS/UserToDev>, archived at: <https://perma.cc/M4B3-JJEA>) to support do-it-yourself customisation.

During a typical FAIMS-led deployment, researchers work with FAIMS Project staff to articulate their data model and workflow. A 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 offer nuanced control the data schema (XML), the user interface (XML and CSS), and automation and logic (Beanshell). The interface can also be translated into multiple languages using a (plain text) localisation file. Completed modules are deployed to a local or online Ubuntu server, and from there onto as many Android devices

88 as needed (after the core mobile application is installed, e.g. from Google
89 Play). Data is then collected using those devices, which can operate fully
90 offline, and synchronised opportunistically when a connection to the server is
91 available. Data can be validated at the time of entry on the device, or later
92 on the server. At the end of data collection, data is exported in the user's
93 desired format by means of a customisable exporter. Three deployment case
94 studies have been published in Sobotkova, et al., 2016[8].

95 Alternatively, FAIMS has developed a XML-based domain specific lan-
96 guage (DSL) to simplify customisation. Using this DSL, a single file is used
97 to generate a complete definition packet, at the expense of some loss of in-
98 dependent control over each element of a customisation (data schema, UI,
99 automation).

100 In addition to deployments conducted by the FAIMS team, projects have
101 independently customised FAIMS Mobile themselves using both the detailed
102 approach of producing an entire definition packet and the simplified DSL-
103 based method[17, 18]. Users who are satisfied with one of the many modules
104 in our GitHub library (<https://github.com/FAIMS>) can also simply instan-
105 tiate an existing customisation.

106 2. Software description

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

115 2.1. Software Architecture

116 FAIMS Mobile consists of 'core' software written in Java and Ruby, cus-
117 tomised to particular field deployments using reusable definition packets con-
118 sisting of XML, Beanshell, and CSS files (which can be generated from a
119 single file written in an XML-based DSL). More specifically, FAIMS uses the
120 following technologies:

- 121 • Javarosa to render native Android UI elements at runtime;
- 122 • Sqlite3 to store an attribute-key-value datastore (with data schemas
123 definable at runtime);

- 124 • An append-only data model inspired by Google’s Protobufs;
- 125 • Beanshell to provide runtime scripting via calls to an underlying Java
126 API;
- 127 • Spatialite to encode geospatial data in the datastore;
- 128 • Nutiteq to render geospatial data;
- 129 • NativeCSS to style android-native elements;
- 130 • Antlr3 as a grammar parser for identifiers; and a
- 131 • A Ruby on Rails / Apache stack to provide a server for synchronisa-
132 tion, version review, user management, and similar tasks, which can be
133 hosted online or on modest hardware in the field.

134 We developed this architecture to meet two fundamental requirements:
135 (1) the software had to accommodate a wide range of research designs, data
136 schemas, and workflows, and (2) the software had to accommodate extremely
137 variable structured, free text, multimedia, and geospatial data. We needed
138 to build a system capable of rendering and recording arbitrary field data,
139 since individual ‘data loggers’ tied to a particular methodologies (even if
140 extensible) and built as separate mobile applications would not appeal to a
141 large enough audience to warrant the investment required.

142 Our Android client can render definition packets at runtime to instanti-
143 ate an arbitrary data collection methodology (schema and workflow), save
144 records to a datastore, and opportunistically synchronize that data with a
145 server and from there to other mobile devices. This distinction between
146 the ‘core’ client and the definition packet resembles the one between a web
147 browser and a website. A browser contains many sophisticated engines for
148 rendering the page, its interactivity, and its styling, but does not have con-
149 tent. A website uses the HTML engine provided by the browser to display
150 its specific content.

151 Four years of deployment experience revealed the importance of quality
152 assurance, something too often neglected in academic software [19, 20]. Each
153 customisation and deployment is, indeed, a miniature software development
154 project [8]. Due to the need for significant QA per deployment, FAIMS Mo-
155 bile 2.5+ supports Robotium for unit and integration tests on customised
156 data collection modules, such that large amounts of test data can be auto-
157 matically added via the normal user interface. This allows users to load-test
158 their modules under simulated field conditions.

159 2.2. *Software Functionalities*

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

- 164 • Deep customisation of data schema, user interface, and automation us-
165 ing either a packet of XML, Beanshell, and CSS documents for nuanced
166 control, or a single file in an XML-based domain-specific language for
167 ease of deployment. Definition document(s) are separate from core
168 software, making modification and reuse easier.
- 169 • Collection of various data types within a single record, including struc-
170 tured data, geospatial data, free text, sensor-produced multimedia, and
171 file attachments.
- 172 • Automated, bidirectional synchronisation of all data across an unlim-
173 ited number of devices using a local or online server. Replication of
174 the entire datastore on each device, not caching, provides robust offline
175 capability.
- 176 • Opportunistic synchronisation whenever a connection is available, al-
177 lowing devices to work in network-degraded environments or offline for
178 extended periods of time.
- 179 • Configurable synchronisation of multimedia files; e.g., to reduce device
180 storage demands, a full-resolution image can be kept on the server while
181 only thumbnails are copied to devices.
- 182 • Defaults, flow logic, hierarchical selections, dynamic UI (expand, col-
183 lapse, hide, or show input fields), and other advanced data collection
184 features.
- 185 • Mobile GIS supporting raster and vector data, layer management, legacy
186 data visualisation, and point, line, and polygon creation and editing.
187 Multiple records can be linked to a single shape, or multiple shapes to
188 a single record.
- 189 • Offline mapping. Base maps and legacy data are uploaded to the server
190 and pushed to all devices. Geospatial data (vectors) created in the field
191 is synchronised across all devices.

- 192 • ‘Annotation’ and ‘certainty’ fields attached to every record. The former
193 allows the collection of granular metadata (mimicking the ‘margins of
194 the page’ in paper recording), while the latter allows users to record
195 their confidence in an observation.
- 196 • Internal and external sensor and peripheral support. All internal device
197 sensors can be called (e.g., camera, microphone, GPS, etc.). External
198 Bluetooth devices like GPS receivers, USB / HID devices like digital
199 balances and calipers, and Bluetooth or USB label printers can be
200 connected.
- 201 • Multilingual support using a plain-text localisation file.
- 202 • An append-only datastore providing a full revision history, including
203 the ability to review and reverse changes to records selectively.
- 204 • Mobile device and server-side validation.
- 205 • Aids to good practice including contextual HTML help, ‘picture dictio-
206 naries’ (selections based on images), and selection trees that can guide
207 users through complex processes.
- 208 • Embedding of URIs into controlled vocabularies or other elements to
209 link them to shared vocabularies, thesauri, or ontologies.
- 210 • Customisable export to desktop software, pre-existing databases, or
211 online data services (using SQL queries).

212 *2.3. Sample code snippets analysis*

213 While a detailed discussion of module code is out of scope for this paper,
214 two documents discuss module creation from start to finish. ‘FAIMS User to
215 Developer Documentation’ (<https://github.com/FAIMS/UserToDev>, archived
216 at: <https://perma.cc/M4B3-JJEA>) is designed to walk users through the
217 creation of a module from first principles. The ‘FAIMS Cookbook’ is a tuto-
218 rial covering data structures and API ([https://faimsproject.atlassian.
219 net/wiki/x/RgAu](https://faimsproject.atlassian.net/wiki/x/RgAu), archived: <https://perma.cc/H6XJ-X6E2>).

220 **3. Illustrative Examples**

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

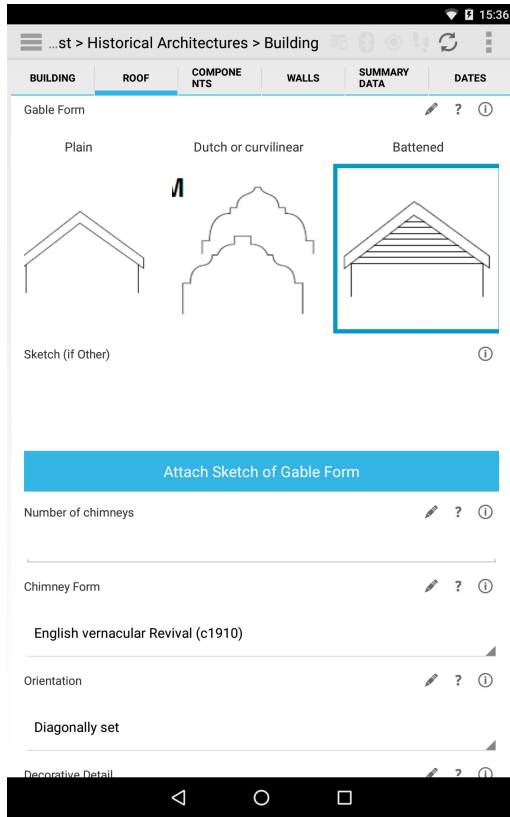


Figure 1: Structured data recording: including dropdowns, numeric fields, checkboxes, radio buttons, and ‘picture dictionaries’

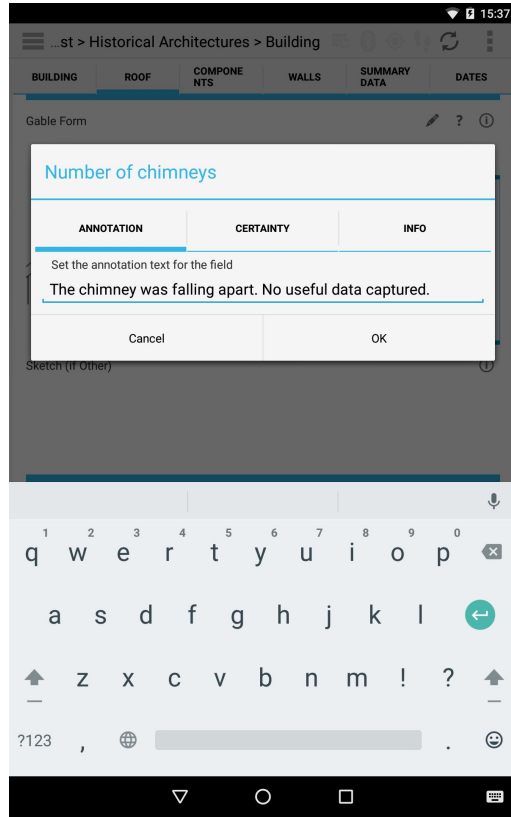


Figure 2: FAIMS has metadata such as annotations (digital ‘scribbling on the margin’) and certainty. Granular, contextualised, HTML-format help (‘info’) is also delivered using this interface

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

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). The software is free (as in speech), customisable, and extensible, accommodating arbitrary research designs. It is also purpose-built for field research. Such research represents only a fraction of the market for consumer or business software, and is unlikely to



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

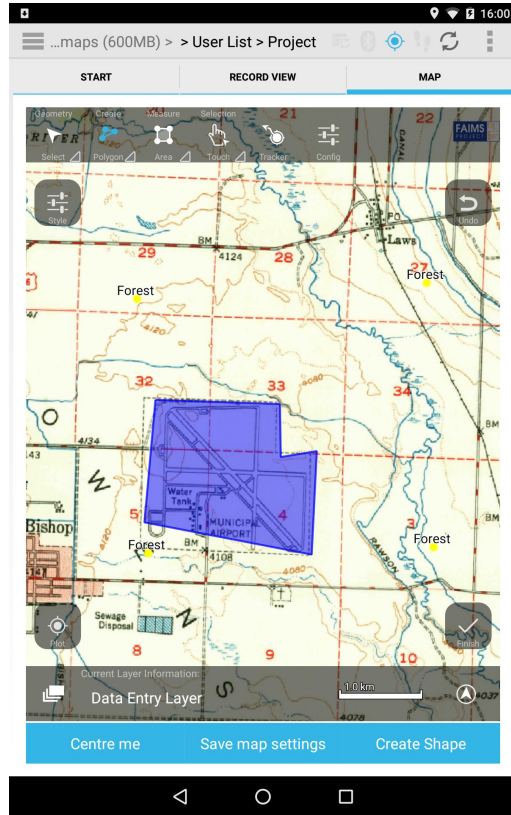


Figure 4: And topo maps

drive its development - whereas it is the sole focus of FAIMS. Finally, FAIMS Mobile is community driven; if current or potential users request new features, they can be implemented (within resource constraints). Researchers can also build and share their own customisations [8], while organisations with sufficient development capacity are welcome to contribute directly to the core software.

Beyond the immediate needs of users, FAIMS Mobile improves research practice and data management. URIs can be embedded in controlled vocabularies and other elements[15], connecting them to linked open data sources (e.g., species information can be linked to the Encyclopedia of Life[21]). Localisation can be used to ‘translate’ a local language of practice to a standard vocabulary (e.g., archaeological ‘context’ or ‘locus’ can be translated to ‘statigraphic unit’ - and then linked to an online ontology). Customisable data export can format data for 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

252 or GeoJSON for ingest into domain-specific repositories like Open Context).
253 Perhaps most importantly, comprehensive, rather than selective, datasets can
254 be created and exported for publication, improving transparency and repro-
255 ducibility. Together, these features that improve data compatibility across
256 projects, facilitating large-scale field research.

257 FAIMS Mobile makes digital recording a more feasible and less costly
258 option for researchers [8, 15]. The core software does the ‘heavy lifting’
259 of field recording (data storage, bi-directional synchronisation, GIS, etc.),
260 and can be customised by leveraging either the control offered by the full
261 definition packet or the efficiency of the DSL module generator. An ex-
262 perience developer can rapidly prototype a recording system if data and
263 workflow models are available (well-scoped systems of moderate complexity
264 can be prototyped in one to two developer-days). Reuse and modification of
265 existing customisations from a growing, openly-licensed online library (lever-
266 aging version control systems like GitHub) also helps to reduce deployment
267 costs[13]. Customisation of FAIMS Mobile is therefore less expensive than
268 production of bespoke mobile applications, and competitive with deploying
269 the suite of generic tools field research requires: a DBMS, GIS, social survey
270 software, multimedia management software, note-taking software, etc. [11].
271 At the same time, FAIMS offers better integration of different data types
272 and requires fewer compromises on the part of the researcher compared to
273 generic tools. Since FAIMS is also easier to redeploy than customised suites
274 of generic tools, it allows practices and innovations to be readily shared[13].

275 FAIMS Mobile has changed users’ daily practice. Three case studies in-
276 volving archaeological deployments [8] indicate that users benefit from the
277 increased efficiency of fieldwork, in that the time saved by avoiding digitisa-
278 tion more than offsets the time required to implement FAIMS and more data
279 collected during fieldwork of a given length. Born-digital data avoided prob-
280 lems with delayed digitisation, which often occurred long after field recording
281 when the context of the record had been forgotten, or the person who made
282 the record was no longer available. Researchers reported more complete, con-
283 sistent, and granular data. They noted that information could be exchanged
284 more quickly between excavators and specialists, which in one case improved
285 ‘post-excavation reconstruction of the site’ and facilitated the evaluation of
286 patterns for meaning in another. They also observed that the process of
287 moving from paper to digital required comprehensive reviews of field prac-
288 tice, during which knowledge implicit in existing systems to become explicit
289 and data was modelled more carefully. By participating in a ‘miniature soft-
290 ware development project’, researchers gained familiarity with the strengths,
291 limits, and demands of software, especially the need for extensive testing.
292 The greatest challenge posed by the transition from paper has been the re-

allocation of time from the end of a project (digitisation) to the beginning (data modelling, development, and testing), even if an overall time-savings is realised.

Although the transition to digital recording during fieldwork represents a significant socio-technical change, FAIMS Mobile has seen good uptake. Since 2013, FAIMS Mobile has been customised at least 40 times (by our team or independently), with 29 confirmed field deployments, nine of which included multiple field seasons (<https://faimsproject.atlassian.net/wiki/x/jBH3B> <https://perma.cc/RW4K-7X2U>). Considering only FAIMS-led development, approximately 300 users have logged 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. Fourteen archaeology, ecology, and history projects are scheduled for 2017 with an estimated usage of another 10,000 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 FAIMS Mobile more attractive to commercial users. This award is also funding community heritage and citizen science deployments, where members of the public can download preconfigured versions of FAIMS Mobile from Google Play (as discrete APKs; See our current set at <https://play.google.com/store/apps/developer?id=FAIMS%20Project> <https://perma.cc/574T-QSVD> which included a Burial Mounds data collector and a ADS Gravestones data collector with screenshots of both) to monitor 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. Doing so often requires several tools (some of which are individually complex) to accommodate the rich and varied data they must collect. FAIMS Mobile offers an alternative. It is 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

332 support, certainties and annotations attached to individual fields, and rich
333 contextual help. It includes customisable export to existing databases or in
334 standard formats, supported by features that facilitate data compatibility.
335 It is designed for rapid prototyping and easy redeployability to reduce the
336 costs of implementation, leveraging online software version control systems
337 like GitHub. FAIMS Mobile is community-driven, customisable, extensible
338 software that can support the socio-technical transition from paper to digital
339 in field research disciplines and facilitate the production of comprehensive,
340 compatible datasets to improve synthetic research, transparency, and repro-
341 ducibility.

342 **Acknowledgements**

343 We would like to thank the many individuals and organisation who have
344 contributed the FAIMS Project since 2012. For a complete list of project
345 sponsors and participants, see <https://www.faims.edu.au/>.

346 Funding: This work was supported by the National eResearch Collabo-
347 ration Tools and Resources programme (RT043; V005), the Australian Re-
348 search Council (LE140100151), the New South Wales Government Research
349 Attraction and Acceleration Program, Macquarie University, Flinders Uni-
350 versity, La Trobe University, Southern Cross University, the University of
351 Queensland, the University of Sydney, UNSW Australia, the Commonwealth
352 Scientific and Industrial Research Organisation ON Prime programme, and
353 fees paid by our users for consultation, customisation, deployment, and sup-
354 port services.

355 **References**

- 356 [1] M. McNutt, K. Lehnert, B. Hanson, B. A. Nosek, A. M. Ellison, J. L.
357 King, Liberating field science samples and data, *Science* 351 (6277)
358 (2016) 1024–1026.
- 359 [2] C. L. Borgman, *Big Data, Little Data, No Data: Scholarship in the*
360 *Networked World*, MIT Press, 2015.
- 361 [3] E. C. Kansa, A. Bissell, Web syndication approaches for sharing primary
362 data in “small science” domains, *Data Science Journal* 9 (2010) 42–53.
- 363 [4] K. Kintigh, The promise and challenge of archaeological data integra-
364 tion, *Am. Antiq.* 71 (3) (2006) 567–578.

- 365 [5] D. R. Snow, M. Gahegan, C. L. Giles, K. G. Hirth, G. R. Milner, P. Mi-
366 tra, J. Z. Wang, Others, Cybertools and archaeology, *Science* 311 (5763)
367 (2006) 958.
- 368 [6] T. Blanke, M. Hedges, A data research infrastructure for the arts and
369 humanities, in: S. C. Lin, E. Yen (Eds.), *Managed Grids and Cloud Sys-*
370 *tems in the Asia-Pacific Research Community*, Springer, Boston, MA,
371 2010, pp. 179–191.
- 372 [7] K. W. Kintigh, J. H. Altschul, M. C. Beaudry, R. D. Drennan, A. P.
373 Kinzig, T. A. Kohler, W. F. Limp, H. D. G. Maschner, W. K. Michener,
374 T. R. Pauketat, P. Peregrine, J. A. Sabloff, T. J. Wilkinson, H. T.
375 Wright, M. A. Zeder, Grand challenges for archaeology, *Proc. Natl.*
376 *Acad. Sci. U. S. A.* 111 (3) (2014) 879–880.
- 377 [8] A. Sobotkova, S. Ross, B. Ballsun-Stanton, A. Fairbairn, J. Thompson,
378 P. VanValkenburgh, Measure twice, cut once: Cooperative deployment
379 of a generalized, Archaeology-Specific field data collection system, in:
380 E. W. Averett, J. M. Gordon, D. B. Counts (Eds.), *Mobilizing the Past*
381 *for a Digital Future; the Potential of Digital Archaeology*, The Digital
382 Press at University of North Dakota, The University of North Dakota,
383 Grand Forks, 2016, pp. 334–370.
- 384 [9] S. B. Fee, Reflections on custom mobile app development for archaeo-
385 logical data collection, in: E. W. Averett, J. M. Gordon, D. B. Counts
386 (Eds.), *Mobilizing the Past: Recent Approaches to Archaeological Field-*
387 *work in the Digital Age*, The Digital Press at University of North
388 Dakota, 2016, pp. 219–234.
- 389 [10] S. J. R. Ellis, Are we ready for new (digital) ways to record archaeo-
390 logical fieldwork? a case study from Pompeii, in: E. W. Averett, J. M.
391 Gordon, D. B. Counts (Eds.), *Mobilizing the Past: Recent Approaches*
392 *to Archaeological Fieldwork in the Digital Age*, The Digital Press at
393 University of North Dakota, 2016, pp. 49–74.
- 394 [11] B. Carter, Crafting work flow- KoBoToolbox/ PostGIS/ QGIS/
395 LibreOffice base/ pgadminIII — institute on digital archaeol-
396 ogy method & practice, [http://digitalarchaeology.msu.edu/](http://digitalarchaeology.msu.edu/crafting-work-flow-kobo-toolbox-postgis-qgis-libreoffice-base-pgadminiii/)
397 [crafting-work-flow-kobo-toolbox-postgis-qgis-libreoffice-base-pgadminiii/](http://digitalarchaeology.msu.edu/crafting-work-flow-kobo-toolbox-postgis-qgis-libreoffice-base-pgadminiii/),
398 archived at <https://perma.cc/5JPH-XQN4>, Accessed: 2017-2-25
399 (2016).

- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] K. Kelly, Better than free, <http://kk.org/thetechnium/better-than-fre/>, accessed: 2017-3-29 (2008).
URL <http://kk.org/thetechnium/better-than-fre/>
- [17] J. Good, KPAAM-CAM overview, in: *KPAAM-CAM – Crossroads Workshop*, 2016, archived at <https://perma.cc/VND8-X2JH>, Accessed: 2017-2-23.
- [18] G. Kiley, Student-developed software streamlines archaeological analysis, <https://news.brown.edu/articles/2016/09/tablet>, archived at <https://perma.cc/8MMX-94K3>, Accessed: 2017-2-23 (2016).
- [19] 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).
- [20] 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).
- [21] E. O. Wilson, The encyclopedia of life, *Trends Ecol. Evol.* 18 (2) (2003) 77–80.

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	<p>Core Application https://github.com/FAIMS/faims-android</p> <p>Server https://github.com/FAIMS/faims-web</p> <p>Definition Packets https://github.com/FAIMS</p>
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	<p>Module Cookbook https://faimsproject.atlassian.net/wiki/x/RgAu</p> <p>Module Beanshell API https://faimsproject.atlassian.net/wiki/display/FAIMS/Program+Logic+Support</p> <p>Developer documentation home https://faimsproject.atlassian.net/wiki/spaces/FAIMS/overview</p> <p>‘User to Developer’ documentation (https://github.com/FAIMS/UserToDev, archived at: https://perma.cc/M4B3-JJEA)</p>
C8	Support email for questions	support@fedarch.org

Table .1: Code metadata (mandatory)

Nr.	(Executable) software meta-data description	Please fill in this column
S1	Current software version	2.5.20
S2	Permanent link to executables of this version	<p>FAIMS Mobile http://www.fedarch.org/apk/</p> <p>Google Play https://play.google.com/store/apps/details?id=au.edu.faims.mq.fieldresearch2&hl=en</p> <p>Server Installer (wget and pipe to bash) https://raw.githubusercontent.com/FAIMS/faims-web/master/installer/puppetInstall.sh</p>
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)