

10-10-2016

## 2.1. Reflections on Custom Mobile App Development for Archaeological Data Collection

Samuel B. Fee

Washington and Jefferson College, samfee@gmail.com

Follow this and additional works at: [https://dc.uwm.edu/arthist\\_mobilizingthepast](https://dc.uwm.edu/arthist_mobilizingthepast)



Part of the [Classical Archaeology and Art History Commons](#)

---

### Recommended Citation

Fee, Samuel B., "Reflections on Custom Mobile App Development for Archaeological Data Collection" In *Mobilizing the Past for a Digital Future: The Potential of Digital Archaeology*, edited by Erin Walcek Averett, Jody Michael Gordon, and Derek B. Counts, 221-236. Grand Forks, ND: The Digital Press at the University of North Dakota, 2016.

This Book is brought to you for free and open access by UWM Digital Commons. It has been accepted for inclusion in Mobilizing the Past by an authorized administrator of UWM Digital Commons. For more information, please contact [open-access@uwm.edu](mailto:open-access@uwm.edu).

# MOBILIZING *the* PAST *for a* DIGITAL FUTURE

The Potential of  
Digital Archaeology



*Edited by*  
Erin Walcek Averett  
Jody Michael Gordon  
Derek B. Counts

MOBILIZING THE PAST  
FOR A  
DIGITAL FUTURE



# MOBILIZING *the* PAST *for a* DIGITAL FUTURE

The Potential of  
Digital Archaeology

*Edited by*

Erin Walcek Averett  
Jody Michael Gordon  
Derek B. Counts



The Digital Press @  
The University of North Dakota  
Grand Forks

Creative Commons License  
This work is licensed under a  
Creative Commons  
By Attribution  
4.0 International License.



2016 The Digital Press @ The University of North Dakota

This offprint is from:

Erin Walcek Averett, Jody Michael Gordon, and Derek B. Counts,  
*Mobilizing the Past for a Digital Future: The Potential of Digital  
Archaeology*. Grand Forks, ND: The Digital Press at the University of  
North Dakota, 2016.

This is the information for the book:

Library of Congress Control Number: 2016917316

The Digital Press at the University of North Dakota, Grand Forks, North  
Dakota

ISBN-13: 978-062790137

ISBN-10: 062790137

# Table of Contents

Preface & Acknowledgments	v
How to Use This Book	xi
Abbreviations	xiii
<b>Introduction</b>	
Mobile Computing in Archaeology: Exploring and Interpreting Current Practices <i>Jody Michael Gordon, Erin Walcek Averett, and Derek B. Counts</i>	1
<b>Part 1: From Trowel to Tablet</b>	
1.1. Why Paperless: Technology and Changes in Archaeological Practice, 1996–2016 <i>John Wallrodt</i>	33
1.2. Are We Ready for New (Digital) Ways to Record Archaeological Fieldwork? A Case Study from Pompeii <i>Steven J.R. Ellis</i>	51
1.3. Sangro Valley and the Five (Paperless) Seasons: Lessons on Building Effective Digital Recording Workflows for Archaeological Fieldwork <i>Christopher F. Motz</i>	77
1.4. DIY Digital Workflows on the Athienou Archaeological Project, Cyprus <i>Jody Michael Gordon, Erin Walcek Averett, Derek B. Counts, Kyosung Koo, and Michael K. Toumazou</i>	111
1.5. Enhancing Archaeological Data Collection and Student Learning with a Mobile Relational Database <i>Rebecca Bria and Kathryn E. DeTore</i>	143

1.6. Digital Archaeology in the Rural Andes: Problems and Prospects <i>Matthew Sayre</i>	183
--	-----

1.7. Digital Pompeii: Dissolving the Fieldwork-Library Research Divide <i>Eric E. Poehler</i>	201
---	-----

## **Part 2: From Dirt to Drones**

2.1. Reflections on Custom Mobile App Development for Archaeological Data Collection <i>Samuel B. Fee</i>	221
---	-----

2.2. The Things We Can Do With Pictures: Image-Based Modeling and Archaeology <i>Brandon R. Olson</i>	237
---	-----

2.3. Beyond the Basemap: Multiscalar Survey through Aerial Photogrammetry in the Andes <i>Steven A. Wernke, Carla Hernández, Giancarlo Marcone, Gabriela Oré, Aurelio Rodríguez, and Abel Traslaviña</i>	251
--	-----

2.4. An ASV (Autonomous Surface Vehicle) for Archaeology: The Pladypos at Caesarea Maritima, Israel <i>Bridget Buxton, Jacob Sharvit, Dror Planer, Nikola Mišković, and John Hale</i>	279
---	-----

## **Part 3: From Stratigraphy to Systems**

3.1. Cástulo in the 21st Century: A Test Site for a New Digital Information System <i>Marcelo Castro López, Francisco Arias de Haro, Libertad Serrano Lara, Ana L. Martínez Carrillo, Manuel Serrano Araque, and Justin St. P. Walsh</i>	319
--	-----



3.2. Measure Twice, Cut Once: Cooperative Deployment of a Generalized, Archaeology-Specific Field Data Collection System <i>Adela Sobotkova, Shawn A. Ross, Brian Ballsun-Stanton, Andrew Fairbairn, Jessica Thompson, and Parker VanValkenburgh</i>	337
3.3. CSS For Success? Some Thoughts on Adapting the Browser-Based Archaeological Recording Kit (ARK) for Mobile Recording <i>J. Andrew Dufton</i>	373
3.4. The Development of the <i>PaleoWay</i> : Digital Workflows in the Context of Archaeological Consulting <i>Matthew Spigelman, Ted Roberts, and Shawn Fehrenbach</i>	399

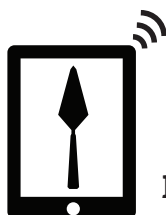
#### **Part 4: From a Paper-based Past to a Paperless Future?**

4.1. Slow Archaeology: Technology, Efficiency, and Archaeological Work <i>William Caraher</i>	421
4.2. Click Here to Save the Past <i>Eric C. Kansa</i>	443

#### **Part 5: From Critique to Manifesto**

5.1. Response: Living a Semi-digital Kinda Life <i>Morag M. Kersel</i>	475
5.2. Response: Mobilizing (Ourselves) for a Critical Digital Archaeology <i>Adam Rabinowitz</i>	493
Author Biographies	521





## Preface & Acknowledgments

This volume stems from the workshop, “Mobilizing the Past for a Digital Future: the Future of Digital Archaeology,” funded by a National Endowment for the Humanities Digital Humanities Start-Up grant (#HD-51851-14), which took place 27-28 February 2015 at Wentworth Institute of Technology in Boston (<http://uwm.edu/mobilizing-the-past/>). The workshop, organized by this volume’s editors, was largely spurred by our own attempts with developing a digital archaeological workflow using mobile tablet computers on the Athienou Archaeological Project (<http://aap.toumazou.org>; Gordon *et al.*, Ch. 1.4) and our concern for what the future of a mobile and digital archaeology might be. Our initial experiments were exciting, challenging, and rewarding; yet, we were also frustrated by the lack of intra-disciplinary discourse between projects utilizing digital approaches to facilitate archaeological data recording and processing.

Based on our experiences, we decided to initiate a dialogue that could inform our own work and be of use to other projects struggling with similar challenges. Hence, the “Mobilizing the Past” workshop concept was born and a range of digital archaeologists, working in private and academic settings in both Old World and New World archaeology, were invited to participate. In addition, a livestream of the workshop allowed the active participation on Twitter from over 21 countries, including 31 US states (@MobileArc15, #MobileArc).<sup>1</sup>

---

<sup>1</sup> For commentary produced by the social media followers for this event, see: <https://twitter.com/electricarchaeo/status/571866193667047424>, <http://shawngraham.github.io/exercise/mobilearcday1wordcloud.html>, <https://twitter.com/electricarchaeo/status/571867092091338752>, <http://www.diachronicdesign.com/blog/2015/02/28/15-mobilizing-the-past-for-the-digital-future-conference-day-1-roundup/>.

Although the workshop was initially aimed at processes of archaeological data recording in the field, it soon became clear that these practices were entangled with larger digital archaeological systems and even socio-economic and ethical concerns. Thus, the final workshop's discursive purview expanded beyond the use of mobile devices in the field to embrace a range of issues currently affecting digital archaeology, which we define as the use of computerized, and especially internet-compatible and portable, tools and systems aimed at facilitating the documentation and interpretation of material culture as well as its publication and dissemination. In total, the workshop included 21 presentations organized into five sessions (see program, <http://mobilizingthepast.mukurtu.net/digital-heritage/mobilizing-past-conference-program>), including a keynote lecture by John Wallrodt on the state of the field, "Why paperless?: Digital Technology and Archaeology," and a plenary lecture by Bernard Frischer, "The Ara Pacis and Montecitorio Obelisk of Augustus: A Simpirical Investigation," which explored how digital data can be transformed into virtual archaeological landscapes.

The session themes were specifically devised to explore how archaeological data was digitally collected, processed, and analyzed as it moved from the trench to the lab to the digital repository. The first session, "App/Database Development and Use for Mobile Computing in Archaeology," included papers primarily focused on software for field recording and spatial visualization. The second session, "Mobile Computing in the Field," assembled a range of presenters whose projects had actively utilized mobile computing devices (such as Apple iPads) for archaeological data recording and was concerned with shedding light on their utility within a range of fieldwork situations. The third session, "Systems for Archaeological Data Management," offered presentations on several types of archaeological workflows that marshal born-digital data from the field to publication, including fully bespoke paperless systems, do-it-yourself ("DIY") paperless systems, and hybrid digital-paper systems. The fourth and final session, "Pedagogy, Data Curation, and Reflection," mainly dealt with teaching digital methodologies and the use of digital repositories and linked open data to enhance field research. This session's final paper, William Caraher's "Toward a Slow Archaeology," however, noted digital archaeology's successes in terms of

time and money saved and the collection of more data, but also called for a more measured consideration of the significant changes that these technologies are having on how archaeologists engage with and interpret archaeological materials.

The workshop's overarching goal was to bring together leading practitioners of digital archaeology in order to discuss the use, creation, and implementation of mobile and digital, or so-called "paperless," archaeological data recording systems. Originally, we hoped to come up with a range of best practices for mobile computing in the field – a manual of sorts – that could be used by newer projects interested in experimenting with digital methods, or even by established projects hoping to revise their digital workflows in order to increase their efficiency or, alternatively, reflect on their utility and ethical implications. Yet, what the workshop ultimately proved is that there are many ways to "do" digital archaeology, and that archaeology as a discipline is engaged in a process of discovering what digital archaeology should (and, perhaps, should not) be as we progress towards a future where all archaeologists, whether they like it or not, must engage with what Steven Ellis has called the "digital filter."

So, (un)fortunately, this volume is not a "how-to" manual. In the end, there seems to be no uniform way to "mobilize the past." Instead, this volume reprises the workshop's presentations—now revised and enriched based on the meeting's debates as well as the editorial and peer review processes—in order to provide archaeologists with an extremely rich, diverse, and reflexive overview of the process of defining what digital archaeology is and what it can and should perhaps be. It also provides two erudite response papers that together form a didactic manifesto aimed at outlining a possible future for digital archaeology that is critical, diverse, data-rich, efficient, open, and most importantly, ethical. If this volume, which we offer both expeditiously and freely, helps make this ethos a reality, we foresee a bright future for mobilizing the past.

\* \* \*

No multifaceted academic endeavor like *Mobilizing the Past* can be realized without the support of a range of institutions and individ-

uals who believe in the organizers' plans and goals. Thus, we would like to thank the following institutions and individuals for their logistical, financial, and academic support in making both the workshop and this volume a reality. First and foremost, we extend our gratitude toward The National Endowment for the Humanities (NEH) for providing us with a Digital Humanities Start-Up Grant (#HD-51851-14), and especially to Jennifer Serventi and Perry Collins for their invaluable assistance through the application process and beyond. Without the financial support from this grant the workshop and this publication would not have been possible. We would also like to thank Susan Alcock (Special Counsel for Institutional Outreach and Engagement, University of Michigan) for supporting our grant application and workshop.

The workshop was graciously hosted by Wentworth Institute of Technology (Boston, MA). For help with hosting we would like to thank in particular Zorica Pantić (President), Russell Pinizzotto (Provost), Charlene Roy (Director of Business Services), Patrick Hafford (Dean, College of Arts and Sciences), Ronald Bernier (Chair, Humanities and Social Sciences), Charles Wiseman (Chair, Computer Science and Networking), Tristan Cary (Manager of User Services, Media Services), and Claudio Santiago (Utility Coordinator, Physical Plant).

Invaluable financial and logistical support was also generously provided by the Department of Fine and Performing Arts and Sponsored Programs Administration at Creighton University (Omaha, NE). In particular, we are grateful to Fred Hanna (Chair, Fine and Performing Arts) and J. Buresh (Program Manager, Fine and Performing Arts), and to Beth Herr (Director, Sponsored Programs Administration) and Barbara Bittner (Senior Communications Management, Sponsored Programs Administration) for assistance managing the NEH grant and more. Additional support was provided by The University of Wisconsin-Milwaukee; in particular, David Clark (Associate Dean, College of Letters and Science), and Kate Negri (Academic Department Assistant, Department of Art History). Further support was provided by Davidson College and, most importantly, we express our gratitude to Michael K. Toumazou (Director, Athienou Archaeological Project) for believing in and supporting our

research and for allowing us to integrate mobile devices and digital workflows in the field.

The workshop itself benefitted from the help of Kathryn Grossman (Massachusetts Institute of Technology) and Tate Paulette (Brown University) for on-site registration and much more. Special thanks goes to Daniel Coslett (University of Washington) for graphic design work for both the workshop materials and this volume. We would also like to thank Scott Moore (Indiana University of Pennsylvania) for managing our workshop social media presence and his support throughout this project from workshop to publication.

This publication was a pleasure to edit, thanks in no small part to Bill Caraher (Director and Publisher, The Digital Press at the University of North Dakota), who provided us with an outstanding collaborative publishing experience. We would also like to thank Jennifer Sacher (Managing Editor, INSTAP Academic Press) for her conscientious copyediting and Brandon Olson for his careful reading of the final proofs. Moreover, we sincerely appreciate the efforts of this volume's anonymous reviewers, who provided detailed, thought-provoking, and timely feedback on the papers; their insights greatly improved this publication. We are also grateful to Michael Ashley and his team at the Center for Digital Archaeology for their help setting up the accompanying Mobilizing the Past Mukurtu site and Kristin M. Woodward of the University of Wisconsin-Milwaukee Libraries for assistance with publishing and archiving this project through UWM Digital Commons. In addition, we are grateful to the volume's two respondents, Morag Kersel (DePaul University) and Adam Rabinowitz (University of Texas at Austin), who generated erudite responses to the chapters in the volume. Last but not least, we owe our gratitude to all of the presenters who attended the workshop in Boston, our audience from the Boston area, and our colleagues on Twitter (and most notably, Shawn Graham of Carlton University for his word clouds) who keenly "tuned in" via the workshop's livestream. Finally, we extend our warmest thanks to the contributors of this volume for their excellent and timely chapters. This volume, of course, would not have been possible without such excellent papers.

As this list of collaborators demonstrates, the discipline of archaeology and its digital future remains a vital area of interest for people who value the past's ability to inform the present, and who

recognize our ethical responsibility to consider technology's role in contemporary society. For our part, we hope that the experiences and issues presented in this volume help to shape new intra-disciplinary and critical ways of mobilizing the past so that human knowledge can continue to develop ethically at the intersection of archaeology and technology.

-----

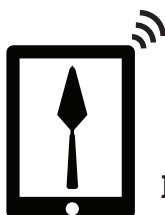
Erin Walcek Averett (Department of Fine and Performing Arts and Classical and Near Eastern Studies, Creighton University)

Jody Michael Gordon (Department of Humanities and Social Sciences, Wentworth Institute of Technology)

Derek B. Counts (Department of Art History, University of Wisconsin-Milwaukee)

October 1, 2016





## How To Use This Book

The Digital Press at the University of North Dakota is a collaborative press and *Mobilizing the Past for a Digital Future* is an open, collaborative project. The synergistic nature of this project manifests itself in the two links that appear in a box at the end of every chapter.

The first link directs the reader to a site dedicated to the book, which is powered and hosted by the Center for Digital Archaeology's (CoDA) Mukurtu.net. The Mukurtu application was designed to help indigenous communities share and manage their cultural heritage, but we have adapted it to share the digital heritage produced at the "Mobilizing the Past" workshop and during the course of making this book. Michael Ashley, the Director of Technology at CoDA, participated in the "Mobilizing the Past" workshop and facilitated our collaboration. The Mukurtu.net site (<https://mobilizingthepast.mukurtu.net>) has space dedicated to every chapter that includes a PDF of the chapter, a video of the paper presented at the workshop, and any supplemental material supplied by the authors. The QR code in the box directs readers to the same space and is designed to streamline the digital integration of the paper book.

The second link in the box provides open access to the individual chapter archived within University of Wisconsin-Milwaukee's installation of Digital Commons, where the entire volume can also be downloaded. Kristin M. Woodward (UWM Libraries) facilitated the creation of these pages and ensured that the book and individual chapters included proper metadata.

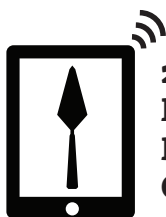
Our hope is that these collaborations, in addition to the open license under which this book is published, expose the book to a wider audience and provide a platform that ensures the continued availability of the digital complements and supplements to the text. Partnerships with CoDA and the University of Wisconsin-Milwaukee reflect the collaborative spirit of The Digital Press, this project, and digital archaeology in general.

## Abbreviations

AAI	Alexandria Archive Institute
AAP	Athienou Archaeological Project
ABS	acrylonitrile butadiene styrene (plastic)
ADS	Archaeological Data Service
Alt-Acs	Alternative Academics
API	application programming interface
ARA	archaeological resource assessment
ARC	Australian Research Council
ARIS	adaptive resolution imaging sonar
ASV	autonomous surface vehicle
BLM	Bureau of Land Management
BLOB	Binary Large Object
BOR	Bureau of Reclamation
BYOD	bring your own device
CAD	computer-aided design
CDL	California Digital Library
CHDK	Canon Hack Development Kit
cm	centimeter/s
CMOS	complementary metal-oxide semiconductor
CoDA	Center for Digital Archaeology
COLLADA	COLLABorative Design Activity
CRM	cultural resource management
CSS	Cascading Style Sheet
CSV	comma separated values
DBMS	desktop database management system
DEM	digital elevation model
DINAA	Digital Index of North American Archaeology
DIY	do-it-yourself
DoD	Department of Defense
DVL	doppler velocity log
EAV	entity-attribute-value
EDM	electronic distance measurement
EU	excavation unit/s
FAIMS	Federated Archaeological Information Management System
fMRI	functional magnetic resonance imaging
GIS	geographical information system
GCP	ground control point
GNSS	global navigation satellite system
GPR	ground-penetrating radar

GUI	graphic user interface
ha	hectare/s
hr	hour/s
Hz	Hertz
HDSM	high-density survey and measurement
ICE	Image Composite Editor (Microsoft)
iOS	iPhone operating system
INS	inertial motion sensor
IPinCH	Intellectual Property in Cultural Heritage
IT	information technology
KAP	Kaymakçı Archaeological Project
KARS	Keos Archaeological Regional Survey
km	kilometer/s
LABUST	Laboratory for Underwater Systems and Technologies (University of Zagreb)
LAN	local area network
LIEF	Linkage Infrastructure Equipment and Facilities
LOD	linked open data
LTE	Long-Term Evolution
m	meter/s
masl	meters above sea level
MEMSAP	Malawi Earlier-Middle Stone Age Project
MOA	memoranda of agreement
MOOC	Massive Online Open Course
NGWSP	Navajo-Gallup Water Supply Project
NeCTAR	National eResearch Collaboration Tools and Resources
NEH	National Endowment for the Humanities
NHPA	National Historic Preservation Act
NPS	National Park Service
NRHP	National Register of Historic Places
NSF	National Science Foundation
OCR	optical character reader
OS	operating system
PA	programmatic agreement
PAP	pole aerial photography
PARP:PS	Pompeii Archaeological Research Project: Porta Stabia
PATA	Proyecto Arqueológico Tuti Antiguo
PBMP	Pompeii Bibliography and Mapping Project
PDA	personal digital assistant

PIARA	Proyecto de Investigación Arqueológico Regional Ancash
PKAP	Pyla-Koutsopetra Archaeological Project
Pladypos	PLAtform for DYnamic POSitioning
PLoS	Public Library of Science
PQP	Pompeii Quadriporticus Project
PZAC	Proyecto Arqueológico Zaña Colonial
QA	quality assurance
QC	quality control
QR	quick response
REVEAL	Reconstruction and Exploratory Visualization: Engineering meets ArchaeoLogy
ROS	robot operating system
ROV	remotely operated vehicle
RRN	Reciprocal Research Network
RSS	Rich Site Summary
RTK	real-time kinetic global navigation satellite system
SfM	structure from motion
SHPO	State Historic Preservation Office
SKAP	Say Kah Archaeological Project
SLAM	simultaneous localization and mapping
SMU	square meter unit/s
SU	stratigraphic unit/s
SVP	Sangro Valley Project
TCP	traditional cultural properties
tDAR	the Digital Archaeological Record
UAV	unmanned aerial vehicle
UNASAM	National University of Ancash, Santiago Antúnez de Mayolo
UQ	University of Queensland
USACE	U.S. Army Corp of Engineers
USBL	ultra-short baseline
USFS	U.S. Forest Service
USV	unmanned surface vehicle
UTM	universal transverse mercator
XML	Extensible Markup Language



## **2.1. Reflections on Custom Mobile App Development for Archaeological Data Collection**

*Samuel B. Fee*

PKapp is a mobile application that facilitates the electronic collection and recording of archaeological field data. Initially implemented during the 2012 season of the Pyla-Koutsopetria Archaeological Project (PKAP), PKapp wedded archaeological methodology with technological innovation (see Bria and DeTore, Ch. 1.5; Ellis, Ch. 1.2; Motz, Ch. 1.3; Poehler, Ch. 1.7). Building on the widespread adoption of tablet computers in 2010, the app turns traditional paper-and-pencil data collection into an electronic process with improved efficiency and speed, which, ultimately, frees up time for researchers to devote to analysis and education.

PKapp was designed as a Web app, rather than a native application. Native apps are written for specific operating systems, whereas Web apps are based on the HTML5 specification. The timing was ripe for developing such an electronic data collection form—HTML5 had become a relatively stable standard in 2011, and mobile computing devices were widespread and inexpensive. From a development standpoint, coding in HTML5 was easier and more reliable than working with earlier, separate versions of HTML and JavaScript (Stark 2010; Stark *et al.* 2012). Also, this approach made it easy to install, test, and operate the software on tablet computers across vast geographic distances—a particularly important point as the developers were in the United States and the archaeologists were in Cyprus.

Tablet computing had quickly been adopted in 2010 for archaeological work (Apple Inc. 2010). The details of that work were already available, making it possible to shape our vision for PKapp from the descriptions of the experience of others (Ellis and Wallrodt 2011). Those early efforts employed apps created by other developers. The

**The Pyla-Koutsopetria Data Collection Form**

Load SU Data
Clear Data / Begin New SU
Record the Data

Area Select One ...

Date DD/MM/YYYY

User Your Name

Supervisor Supervisor

EU # Excavation Unit

SU # Stratigraphic Unit

**Summary**

1-2 Sentence Description

**Description**

Full Description

**Elevations**

Min:  Max:

**Harris Matrix**

This SU is ...

SU #

This SU is ...

SU #

This SU is ...

SU #

**Soil Description**

Munsell Value (wet):

Soil Type: Select One ...

Sediment Consolidation: Select One ...

Stoniness %: Select One ...

Dominant Clast Size: Select One ...

**Features/Photos/Drawings**

+ Features of this SU

+ Photos of this SU

+ Drawings of this SU

**Procedure**

Sieve: Dry

# of bags (wet) Number

**Finds (# of bags)**

Ceramic 0

Stone 0

Bone 0

Metal 0

Organic 0

Plaster 0

Other Enter Description

The Pyla-Koutsopetria Archaeological Project

**Figure 1:** The PKapp mobile app.

development of PKapp was an effort to explore the possibilities of custom software development. In the end, and most importantly, PKapp taught us how to write software for mobile devices while also illuminating numerous possibilities for digital workflow in field research.

The uses for the app have been detailed in a brief article that William Caraher, David Pettegrew, and I composed for *Near Eastern Archeology* (Fee et al. 2013). During the 2012 field season, Caraher and Pettegrew were co-directors for the project along with R. Scott Moore. Caraher served also as database administrator, Pettegrew served as Field Director, and I was in charge of software development. The purpose of this chapter is to describe the technical planning and development behind the app, identify some of the most challenging programming problems we encountered, and suggest current directions for app development given the rapid advance of programing libraries and frameworks (tools that make it easier and faster to develop an application like PKapp today than it was in 2012) for custom mobile app development.

#### DESCRIPTION OF THE APP

PKapp represents a natural progression from traditional paper collection forms, replacing a two-page paper document with a single electronic form for recording basic, required information and unstructured descriptions (FIG. 1). The basic unit of excavation at PKAP is the stratigraphic unit (SU), and thus the entire electronic form is constructed around recording or recalling data for each SU.

As we began planning the project in 2012, we identified a number of parameters that needed to be addressed carefully during the development process:

1. There could be no data loss.
2. Data entry should follow a simple process.
3. Data validation was imperative.
4. The software must run locally on the device (without Internet access).
5. A simple data export mechanism was required.
6. Updates should be accessible remotely.
7. The software must be platform-agnostic, and must run on any mobile device.



We returned frequently to this list in our planning of both design and programming elements (such as the export of data). Several of the criteria, which resulted from the needs of researchers working in remote locations with unreliable internet access, had some technical implications for our work. We worked with the form validation abilities built into HTML5 to ensure that any data entered was of the right type before it ever got to the primary database. We also ensured that the app would write data directly to the device without wireless access, and that it would upload data from the device to the primary database easily—a task easier to theorize than to implement.

Finally, our desire to access updates remotely meant we needed to develop a Web app for use outside of the app-store environment. With such an approach, we could continue to test and revise while working in the field. We could post new versions of the software overnight and have them in use in the field the next day, which would not have been possible with the current app-store distribution model that requires a lengthy approval process. Because we were avoiding app-store distribution and developing a stand-alone Web app, we could embrace fully the open-source standards of HTML5 and ensure that PKapp would run on any device with a stable and current Web browser.

### *App Design*

As mentioned previously, the paper form for recording the field data at PKAP was composed of two pages. The first page asked the recorder to write down information about the context, including name and identifiers (date, supervisor, recorder), location (area, excavation unit, elevation, stratigraphic relationships, universal transverse mercator (UTM) coordinates), soil descriptors (soil type, clast size, Munsell color), associated data (features and photographs), method, and relative quantity of finds by bag. The second page contained identifying fields in case that page became separated from the first, with blank lines for narrative description and interpretation of the area.

With multiple excavators working on site, a major advantage of the digital form is that it forces the recorder to enter data in standardized ways (see Bria and DeTore, Ch. 1.5; Ellis, Ch. 1.2). Some fields require the user to choose from selectable menus, ensuring more normalized data, while in most other data entry locations the user can only

enter specific type of information that actually fits the way the data is tracked in the database. For instance, since the excavation unit (EU) numbers are only two digits—the user cannot enter any more than two into that field. The same holds true for SU numbers, elevations, or any text entry area within the form. The app thus guarantees that the data is formatted in a way that will import directly and correctly into the primary database.

Another PKapp feature that helps with data validation is the ability to bring up the correct numeric or alphabetic keyboard for specific entry fields, thereby reducing the number of button clicks and saving time overall (Clark 2010). This can be done through the use of regular expressions. Regular expression attributes in HTML5, which were most commonly used in the past to evoke pattern matching for searches, allow the software to check the value of the pattern attribute against a regular expression to see if it is valid or not. For instance, this expression:

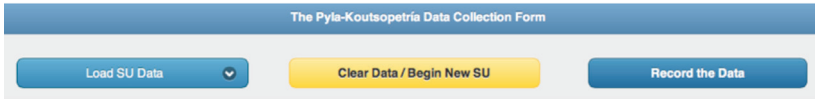
pattern="[0-9]\*"

included as an attribute to the input element would limit the input to numeric values. If it is valid, the form submits; if it is not, the user is asked to correct the format of the entry. Thus, in addition to bringing up the right keyboard in the app, regular expressions give us another means to ensure data validation.

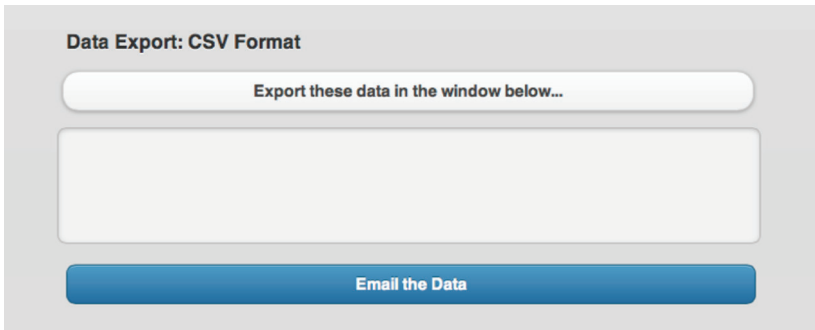
In addition to the above features, there are buttons that facilitate interaction. These buttons enable the primary functions for interacting with the app, and they are also used to access data export functions, which enable the app's data to be exported and later incorporated into the primary database.

### *Interacting with PKapp*

The buttons at the top of the application allow the user not only to enter data correctly, but also to interact with the data that is already stored locally on the device (FIG. 2). For data collection purposes, the stratigraphic unit, which is the primary method of identification for records for fieldwork at PKAP, was used as the unique identifier for the local database.



**Figure 2:** Interacting with the data on the device.



**Figure 3:** Exporting the data.

From the top left, the “Load SU Data” button loads any previously entered SU data. Because PKapp takes advantage of the local storage on the device, a user may view and edit the previously collected data. In essence this function is similar to auto-completion on Web forms through PHP, except that it is loaded from the local database rather than a remote server.

Located in the center, the “Clear Data/Begin New SU” button removes data from the form so the user can enter new data, though previous data can always be re-loaded using the “Load SU Data” button.

The “Record the Data” button writes the data to the local SQL database. This feature is similar to a “Submit” button, but it is modified with specific scripts that execute additional functions, which are discussed below in the “technical difficulties” section.

The remaining interface elements within PKapp allow for the export of data. The “Data Export” section at the bottom of the form contains two buttons and a text field that serve as a window for viewing the data (FIG. 3). The upper button exports the data on the device into CSV (comma-separated version) format and displays those data in the associated window (CSV is a simple, tab-delimited plain-text format that is easily imported into almost any database). This enables users of the app the opportunity to review and validate the data once again before sending it to the database administrator for incorporation into the primary database. The lower button, “Email the Data,” simply emails the data directly to a unique address that has been established for receiving these data for PKAP.

## TECHNICAL DIFFICULTIES

Creating PKapp was especially challenging because we were implementing an innovative but immature toolset—specifically, HTML5 on newer versions of mobile browsers. The HTML5 specification is a collection of HTML, CSS, and JavaScript along with a much more robust support for Web forms. In many ways, this makes it perfect for what we intended with PKapp: a Web app that could be easily and remotely updated even while being deployed in the field. The app therefore consisted of highly customized HTML5, along with the jQuery Mobile library, and specifically the jQuery Mobile JavaScript libraries that handled a lot of the look-and-feel of the app. The customizations made to the library included the additions of form mark-up and a number of

attributes to help validate the data and eliminate a number of potential user errors in the input of data. For the most part, this was all straightforward, and creating this type of app was relatively easy. There were, however, three significant problems with the software that needed to be addressed during our development process.

1. *Features we wanted but could not provide.* We would have liked the app to have the ability to capture photos and attach them to the exact data record for the SU being recorded and to record GPS coordinates for the areas under observation. We simply could not implement these features in 2012 because the application programming interface (API)—code instructions that link into preexisting programs or hardware controls—for the internal camera and GPS were not reliable. Today such APIs, which enable us to make use of certain hardware features we could not otherwise access without developing a native app, are widely available, and these capabilities could be incorporated within PKapp.
2. *The database.* Our local storage on the device consisted of a WebSQL database implemented through JavaScript. It was a challenge to decide which database model to implement since WebSQL had already been deprecated from the HTML5 specification despite the fact that the HTML5 spec had only been published the previous year. (Deprecated elements are removed from the specification and no longer considered “valid”). The alternatives were localStorage, which was being used to save data for the current form so it could not be lost before being saved, and IndexedDB, which unfortunately still was not fully implemented in WebKit browsers such as Google Chrome or Apple Safari. Since WebSQL was deprecated, support and documentation were very limited. This made the implementation of a stable database harder to accomplish. The actual saving of the data simply required a basic understanding of SQL—that itself was not very difficult—but getting the data out of the database in CSV format or back into PKapp for viewing was more challenging.
3. *Exporting the data.* Given that the app was designed with HTML5, we faced an additional problem in that WebKit browsers had not implemented the FileSystem API at the time of development. This meant that the app could not simply write data files and access them later. This then created hurdles in exporting the data, which were circumvented by sending the data to the screen, then using

a separate function to access a remote PHP script to send the data via email. Obviously, this last function only operates when Internet connectivity is present. But this functionality enabled users to review the data locally even if they did not have access to the remote database server.

By far the biggest problem of the three articulated above concerned the transfer of data. Had a reliable form of wireless communication been available, the simpler solution would have been to send the data directly to a PHP script and import it into any SQL server on the Internet. Yet our software solution had to run locally as there was no wireless connectivity at the site at Pyla-Koutsopetria. Thus PKapp needed to be able to view the data locally and send it out when the Internet was accessible. To the best of my knowledge, the process of taking data from `localStorage`, placing it into the app, exporting it into an email, and sending it onward is an approach that had not been tried before.

Another development option would have been to write the app natively as an iOS and/or Android application. Such an approach would have avoided the challenge with data export, and it would have enabled our implementation of local files. But this would have conflicted with our desire to remain platform agnostic and accessible on any mobile device. A native app approach could have also allowed us to work with the Dropbox API, making storage easier and allowing for replication of data when connection was restored. But in order for us to update the app overnight, a native app could not be used without numerous complications for the researchers collecting data in the field.

#### REFLECTIONS ON AND FUTURE POSSIBILITIES FOR CUSTOM MOBILE APP DEVELOPMENT

There were different approaches to writing the software for the application development process, each with their own pluses and minuses (Koch 2014). This underscores the importance of developing a vision for the project at the outset, before sitting down to write any code. Had we not collectively held that vision, we could have easily gone astray at several development stages and ended up with an app that did not address all of the issues that we felt were important to the project.

Because the technological toolset itself was changing even as we were developing PKapp, it would have been easy to change direction at several points—but implementing any of those new tools might have brought innovation in one regard at the expense of another, or even the entire project. And such technological change has only accelerated since 2012.

In 2012 we wrote PKapp with a text editor, various browser software, and the jQuery Mobile framework. An alternative approach could have incorporated so-called off-the-shelf software; indeed, several other projects described in this volume very successfully took that approach (see Gordon *et al.*, Ch. 1.4; Bria and DeTore, Ch. 1.5; Ellis, Ch. 1.2; Motz, Ch. 1.3). But we wanted the control afforded by creating our own custom app. At that time, writing the code manually was the only viable way to accomplish our end by developing a Web form that would operate effectively on a mobile device (Wroblewski 2011). Today there are many tools available for making that process both simpler and more direct, and many of the technical difficulties we faced in 2012 have subsequently been addressed through the release of more formalized JavaScript APIs that now provide access to additional hardware in mobile devices. Finally, the simple maturation of HTML5 has brought about increased stability for the local storage of data within the browser that provides additional reliability for the app itself and confidence in the data integrity of the content that we receive from the device.

One of the core features of HTML5 is the improved handling of forms. Prior to HTML5, expanding form functionality (particularly with data validation) required extensive and often problematic JavaScript programming. With the incorporation of regular expressions into the HTML5 specification, this is now a feature provided through the simple addition of attributes to the form elements. Because PKapp is essentially a data collection form, this aided our development immensely. In addition, the development of JavaScript frameworks and libraries in recent years has made more of the development work we undertook in the past easier today.

### *JavaScript Frameworks*

While libraries, or collections of code available for integration into new programs, typically perform a specific but limited function, frameworks refer to a larger structure—a collection of existing libraries, or scripts, or code that can be utilized to create custom programs. While there are many new JavaScript libraries and frameworks today, we found the jQuery Mobile framework was the best option at the time of development. It was particularly well suited for handling Web forms and all of the components we would likely want for a custom field-data collection tool (items such as selection menus, toggle switches, text entry areas, checkboxes, and the like). New tools for prototyping or further developing jQuery Mobile based apps mean that not everything must be coded manually, nor must all the hooks into the framework be created through a text editor. Software now enables anyone with minimal coding experience to build, at the very least, the front-end of a Web app. This places the design of any custom data collection app firmly within the hands of the archaeologist, and not necessarily a programmer.

These tools come with different approaches and business models. Some are drag-and-drop, others are WYSIWYG (“what you see is what you get”); some are free, yet others are provided at considerable cost. Codiqa is a preferred option. It is available in online and desktop versions, and is free for academic use; however, a \$79 desktop version enables you to keep local control of your files, which is something that is important for any developer. Codiqa exports the HTML, CSS, and JavaScript that is needed to build an app.

Once these files are created, building the front end of the app involves simply modifying and customizing the appearance (via CSS). To create a custom field-data collection tool, one need only to add in the regular expressions to reinforce data validity, set up the local database, and develop an export feature. Some newer JavaScript APIs can further enhance the feature set of the app as described in the next section.

### *JavaScript APIs*

Since we wrote PKapp, two APIs were released that are of particular interest to archaeologists: the camera API and the geolocation API,



two features we wanted but could not provide (as noted above). The camera API allows you to take a picture with your device's camera and load it to the current page. The geolocation API provides the location of the device to the app. These APIs enable the building of a more robust app than we could manage in 2012 with PKapp, though current support for various browsers is still mixed. Nonetheless, these represent the future capabilities for custom data collection apps, so exploring their potential is worth the effort.

There are two caveats to keep in mind with both of these APIs. First, the camera API places an image into the app, then saves it to the database (assuming the database can accept image files). Image files will be large, so the time required for uploading the data to the primary database will become correspondingly significant and the overall size of the database will swell. In fact, most databases contain a data type known as a BLOB (Binary Large Object) just for such use, but this slows the process of data transfer. Second, the geolocation API defaults to a very imprecise setting. When a mobile device cannot quickly acquire a GPS signal, the default settings of the API try to specify location based on Wi-Fi signal or IP address instead. Obtaining good coordinates will require some programming work as well as a recognition that the implementation of this feature will slow down the app, and acquiring good data for location will also likely require connection to a cellular network. In the end, incorporating these APIs will likely require more than a basic knowledge of HTML, but a non-programmer with some considerable skill in HTML5 could complete such a project.

### *Database Advances*

When the HTML5 specification was released in 2010 (although not “officially” released until 2014), there were three approaches to handling client-side databases: `localStorage`, `IndexedDB`, and `WebSQL`. The first, `localStorage`, was problematic in that it does not always indicate when the stage of insufficient storage is reached, which raises the potential for data loss. The second, `IndexedDB`, was not yet recognized by browsers and could not be implemented at the time. Therefore, we chose the third option, `WebSQL`—the most broadly used implementation for databases in most browsers—in spite of the fact that it had already been terminated in 2011. At the same time, because it was still

fully functional in programs like Apple Safari and Google Chrome, we decided it was our best option and chose to move forward.

Today, the choices are largely the same, but browser support is greatly improved. IndexedDB is now supported in Google Chrome and iOS 8, which means that programs using this technology will continue to be supported on browsers in the future. Fortunately, there are even JavaScript libraries that will provide WebSQL translation for older browsers (iOS before version 8). This means that you can count on the work you do today to be relevant in the future.

The primary benefit of the changes over the past few years is that the future direction for development is clear, and those creating apps now do not need to be concerned with issues of obsolescence. Also, more developers are approaching their projects through the use of IndexedDB, and as a result, online resources and information can assist with the development of apps that incorporate IndexedDB storage. Nonetheless, the entire database backend of any custom data collection app is fraught with technical problems. This could very well be the most technically complex aspect of the development project. These difficulties revolve around the challenges of selecting the right database approach and the lack of documentation available for such work.

For those seeking to develop a similar app today, the recommended approach is to utilize IndexedDB while also including a JavaScript library to provide backward compatibility for browsers with WebSQL support. This would give the app a much broader reach in terms of supported devices, and it would also ensure the relevancy of the approach to the local database into the future.

### *Export Problems*

Despite the advances of the past few years, data export remains a difficult conundrum for anyone developing a custom app designed to run without connectivity. Apple has not implemented the `fileSystem` API to help address this issue, but there are other good approaches that simply require some work. For PKapp we exported the data and emailed it so that we could provide another check on the data before incorporating it into the primary database. Today, many other “to-do list” and note-taking apps provide such functionality through Dropbox or other similar cloud-based services. Use of a Dropbox account and

the Dropbox API may be a particularly attractive option for any apps currently being developed.

Of course, should a project enjoy reliable connectivity—even occasionally—an app could be created that simply sends the data to a primary database on a server when connected to the Internet. Since each entry could be given a unique timestamp, entries could be searched daily to verify data integrity. In such a circumstance, data transfer becomes a very smooth operation that risks few technical problems.

In the end both of these solutions are simpler than the one we implemented for PKapp in 2012. With reliable connectivity, an app could possess a richer feature set in this regard than an app designed to work exclusively offline.

## CONCLUSIONS

The development of PKapp taught us a number of important lessons about implementing mobile apps for data collection in archaeological fieldwork. In their simplest forms, mobile apps are not difficult to create—a simple one can be built based upon an RSS feed in minutes. But when considering the collection, storage, and access of data specific to the PKAP project, there were no pre-existing commercial tools that could accomplish our goals. In the end we implemented an app written with HTML5 and some custom JavaScript coding.

Native apps are written for specific operating systems. Web apps are based on the HTML5 specification. We decided on a Web app approach so that we could update the app at any time and post it online for the team to install in Cyprus almost instantaneously. We could fix bugs as they appeared, or modify features based upon actual field use. We thus could actively address our design parameters, which called for easy and quick updating of the software. We also avoided having to write the app for multiple platforms and getting each app and each update approved for delivery through its respective app store.

The Web app development process is even easier today as a host of new tools exist to facilitate such projects. In addition to a number of JavaScript libraries, frameworks, and APIs, there are a plethora of tools such as Codiqa to aid the actual development of the front-end of an app built with HTML5. The ease-of-use present in these tools means that the archaeologist can be actively engaged in the development of

the app, and the software development process becomes truly participatory. With these tools technical support is needed primarily for the development of the local database and the eventual communication with the primary database, wherever it may reside.

In the end, collecting data via PKapp was easy and the app worked remarkably well, matching our design parameters and meeting all of our fieldwork goals. As a result of our experience using the app successfully, we see benefits in the incorporation of mobile technologies for collecting data in the field. There are significant improvements in efficiency and overall time saved, because entire steps in the older process—particularly the manual process of completing paper forms, converting that data into electronic format, and reviewing the resulting electronic data—can be streamlined. The ability to incorporate automatic data validation into the entry process also makes this approach an improvement over traditional methods, which required additional manual validation. This is not to say that such technical efficiencies do not come without a cost (Caraher 2013). Indeed, any field team should weigh the benefits of efficiency as they reflect upon *where* and *when* the analysis and interpretation occurs in the archaeological process for the project.

But a season of testing provided us with enough observation for our data integrity concerns that we have great confidence in the quality of data collected via PKapp. With the advancements and implementation of the HTML5 specification, as well as broader implementation of JavaScript APIs, we could today even more easily produce Web apps for field data collection that run without connectivity. Consequently, this process is increasingly accessible to most researchers, and it seems worthy of consideration for most projects.



<https://mobilizingthepast.mukurtu.net/collection/21-reflections-custom-mobile-app-development-archaeological-data-collection>

[http://dc.uwm.edu/arhist\\_mobilizingthepast/10](http://dc.uwm.edu/arhist_mobilizingthepast/10)

## REFERENCES

- Apple Inc. 2010. "Discovering Ancient Pompeii with iPad." <http://www.apple.com/ipad/pompeii>
- Caraher, W. R. 2013. "Slow Archaeology," *North Dakota Quarterly* 80: 43–52.
- Clark, J. 2010. *Tapworthy: Designing Great iPhone Apps*. Sebastopol, CA: O'Reilly Media.
- Ellis, S. J. R., and J. Wallrodt. 2011. "iPads at Pompeii." *Pompeii Archaeological Research Project: Porta Stabia*, <http://classics.uc.edu/pompeii/index.php/news/1-latest/142-ipads2010.html>
- Fee, S. B., D. K. Pettegrew, and W. R. Caraher. 2013. "Taking Mobile Computing to the Field," *Near Eastern Archaeology* 76: 50–55.
- Koch, P. 2014. *The Mobile Web Handbook*. Freiburg: Smashing Magazine GmbH.
- Stark, J. 2010. *Building iPhone Apps with HTML, CSS, and JavaScript*. Sebastopol, CA: O'Reilly Media.
- Stark, J., B. Jepson, and B. MacDonald. 2012. *Building Android Apps with HTML, CSS, and JavaScript*. 2nd edn. Sebastopol, CA: O'Reilly Media.
- Wroblewski, L. 2011. *Mobile First*. New York: A Book Apart.