CHAPTER

8

Open Formats for Mobile Learning

Geoff Stead

Abstract

Successful mobile learning (mLearning) initiatives are surprisingly diverse. Some prescribe standardised devices and applications to a cohort of learners while others adopt a Bring Your Own Device (BYOD) strategy, using a mesh of different devices, apps and content. Some weave tools into existing learning scenarios, while others use mLearning to create new ones. This diversity creates a challenge for those evaluating emerging technologies as tools for learning (how to identify the specific impact that one piece of technology has on the learning process and outcomes), as well as those hoping to transition good mobile content from one learning scenario to another.

Mobile learning frameworks like FRAME highlight this inter-relationship, showing how the learning outcomes emerge from an interaction between the technology, the learner and the context. However, evaluating mLearning experiences can be difficult because of the many variables.

This chapter builds on research funded by the U.S. government in the Mobile Learning Environment (MoLE) project, which explored the different technologies that underpin most mLearning content, trying to answer the question "Which formats and technologies are best for simplifying the process of moving good mobile learning between different platforms?" The project team researched and tested the most relevant guidelines and standards, as well as building many different prototypes, culminating in a 24-nation live trial on tablets, media players and smartphones. The optimal solution was found to be a "mesh" of different technical approaches, bridging some of the gaps between mobile platforms, and improving the portability of the learning apps that run on them.

The standards and technical approach proposed in this chapter form the basis of the open source OMLET framework (Open Mobile Learning Toolkit). OMLET

is a core component of several commercial deployments in the U.S. and the UK. Content suppliers are starting to adopt these standards to build new mLearning content. Most significantly, the U.S. government JKO (Joint Knowledge Online) platform has adopted them for all future content that they commission.

Introduction

Mobile learning (mLearning) offers many potential benefits to work-based learners. These benefits include the ability to engage in ongoing, professional development during "stolen moments" anytime and anywhere; blending together access to reference materials, performance support and professional development; and 24-hour "just in time" support for immediate needs — all available through the learners' familiar simplicity of their own, personal smartphone. The benefits for employers are also clear, such as an engaged and improving workforce, quicker methods for sharing time-critical data, enhanced access to feedback from employees, and reduced costs of downtime caused by attendance on conventional courses.

These are the accepted wisdoms, but what is really happening? A recent Good Technology report into Bring Your Own Device (BYOD); using employees own mobile devices in the workplace) found that over 75% of employers are already supporting mobile access to the workplace, or plan to within the next year (Good Technology, 2012). Studies looking at mLearning adoption found that a surprising number of employers were planning to use these same devices for accessing learning and training. For example, ASTD estimated 28% in their research paper (ASTD, 2012).

Several studies found that the main barriers to wider corporate adoption were technical (ASTD, 2012; Towards Maturity, 2013). In BYOD scenarios, some devices will be more fit-for-purpose than others, leading to potential lack of parity amongst learners. Supporting a diverse range of devices safely and fairly requires:

- addressing challenges in the deploying, sharing and managing of apps and content (operational),
- managing risks to confidentiality and Intellectual Property Rights (security of data),
- managing access to materials both on- and off-line (variations in Internet access and device capacity), and
- meaningfully tracking usage and measuring progress (analytics/tracking).

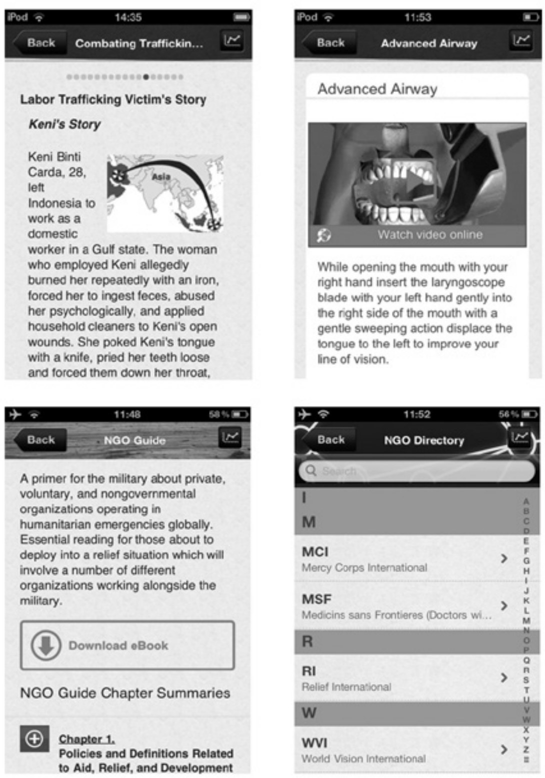
As an investment in the development of effective practice in mLearning, the U.S. government sponsored a two-year technology research project to explore the technical challenges involved in deploying mLearning as a core element of its mainstream eLearning delivery. The Mobile Learning Environment project (MoLE; www.mole-project.net) developed sample content, tools and platforms for mobile, work-based learners involved in humanitarian and disaster relief work. These were deployed under the name Global MedAid, a mobile app for both iOS and Android that was trialled in 24 nations by 270 learners using several language versions (Figure 8.1).

The technology team carried out practical research and prototype development to explore the underpinning technologies required to deliver meaningful mLearning tools and content to a massively diverse user group whose common feature was that they were all very mobile, work-based learners. It became clear that they all shared certain specific requirements of mLearning: they needed small, easy-to-access nuggets of learning mixed with support tools that were quick to locate and easy to use across a wide range of devices. Work-based learning tends to be multi-episodic, informal and just-in-time. Although mobile devices have been shown to foster situated approaches to learning in and across work contexts (Pachler, Pimmer, & Seipold, 2011), the employers of these particular learners had previously forbidden this. In addition, the identified user group was likely to have only occasional access to the Internet, and needed a diverse mix of resources including:

- compliance-based "courses" that require tracking,
- video interviews with domain experts for guidance and support,
- active "checklists" as performance support tools,
- published e-books and other existing resources, and
- mobile reference tools and look-up charts.

Figure 8.1 shows a selection of these screens from the Global MedAid app.

Figure 8.1: Range of different types of content from the Global MedAid app.



The research team worked within this range of resource types, exploring different technical approaches that would allow them to "travel well" between platforms. This was done in reference to traditional eLearning standards, emerging Web, and mobile media trends. The final combination of frameworks and technical standards adopted worked well on a diverse range of devices, and can be used for a wide range of work-based mobile learners. This mesh of standards, though still evolving, has already been adopted by the leading U.S. government eLearning platform as part of a move to mobile. This chapter sets out the basic foundations, shares technical lessons learned, and outlines the proposed open formats.

Technology as an Integral Dimension

Successful mLearning is a complex blend of the learner's own skills, the affordances of the device, the appropriateness of the content, the mix of media, the context of the learning, the fluidity of the software, and the performance of the mobile app itself (Koole 2009; Stead 2012a, 2102b). It is a vast subject area, so this chapter focuses on the technologies that support the links between elements of the overall blend. It looks at content issues (e.g., data formats); the technologies required to create interactivities inside the content; interface design and how this differs across mobile platforms; protocols for sharing packages of mobile content between phones; and mechanisms to share tracking data with learning platforms.

In evaluating mLearning content from this perspective, it is beneficial to include as many of the dimensions as possible, both technical and non-technical. Examples include:

- ease of navigation,
- quality of interactivities and appropriateness for the device in question,
- "findability" of content on a specific device,
- range of supported devices, and
- effort required to move content onto a new mobile device.

These provide challenges for all stakeholders. Some typical dilemmas for mobile developers include:

- For optimum user experience, an app should be developed to target a specific mobile platform (e.g., iOS), but for maximum portability of the content it should not.
- For maximum portability of content, the best technical solution is to use a Web app (hosted online), but this excludes all the best phone features (native menus, camera, GPS, other apps, etc.).
- For the best mLearning experience, users need to be able to work offline, but for integration with traditional eLearning systems, the information needs to sync online.

The solution to these technical dilemmas is to recognise the connections between content, interactivity, the app features and the type of learning itself. The role of the technologist is to try to extract open, re-useable formats and standards that allow these different dimensions to travel well across and between different learners and different platforms.

Good practice in traditional eLearning does not assume that one course or programme will provide comprehensive, one-size-fits-all coverage, despite the claims of some providers. The same applies to good mLearning models. It is unrealistic to expect one single app or course to provide the entire breadth of learning required for any given domain area. The assumption is that a small amount of well-contextualised and focused information is more effective for learning than a large-scale, more ambitious and conventionally structured programme; and that part of the benefit of mobile access is being able to jump into small nuggets of relevant and sometimes personalised information (see Sharples, Taylor, & Vavoula [2005] for a typical example, or Smith [2011] on context awareness). This makes it all the more relevant to support the development of larger pools of small learning nuggets that can be seamlessly assembled in different combinations, for different learners, on different devices, and with links leading outside the mobile world where needed.

Technical Approaches to Sharing Mobile Learning

For technologists developing mLearning apps that they would like to share between multiple device types and multiple potential authors, there are four main approaches to technical development:

- Open apps – Involves software techniques that enable developers to create an app that runs on different mobile phone platforms in a single build (cross-platform development)
- Open content/content formats – Allows individual pieces of content to display on multiple devices, using industry standard "players" (e.g., e-book readers) or with native device support (e.g., audio and video files)
- Open content with embedded interactivity – A hybrid between the above two approaches, and is the ideal scenario for learning interactivities, because it combines content with appropriate learning interactions (e.g., a Web app: HTML+ JavaScript)
- Open protocols, and formats to encourage sharing – Involves both the sharing of content and the sharing of tracking, progress and messaging between applications

All of these were explored during the technical development of the MoLE project, and a combination of the three was found to be the most successful. Details about the optimal formats are described below.

Open Apps: Cross-Platform Application Development

Despite the efforts of many technology experts and enthusiasts engaging in this area, there remains no perfect solution to this challenge of developing a mobile application that will work perfectly on all devices. Each distinct approach has its own merits, and disadvantages, with many mobile developers still believing that there is no viable alternative to native coding for each device type. At the highest level, the main approaches include:

- cross-compilation (code once, but compile multiple times, for different devices),
- mobile Web apps (the app runs in the mobile browser), and
- hybrid Web app (native "player" app, with Web app type content).

These are true of all mobile development, but an open mLearning solution imposes some additional requirements that help to narrow the alternatives:

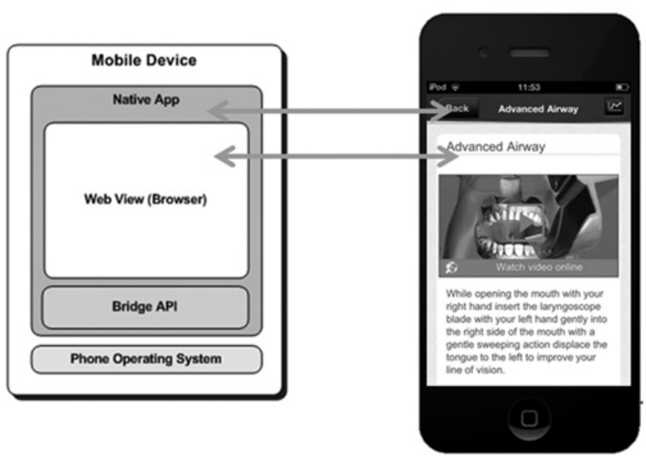
- separation between content (easy to make and share) and platform (needing technical expertise);

- provision of many content types needing embedded interactivity — presentation media alone is not enough;
- offline access to content, but online syncing and tracking;
- deeper integration with underlying mobile features (e.g., GPS, sync, camera, microphone, e-book reader app); and
- maintenance of a familiar user interface, appropriate for whichever device the learner was using.

The project team did a detailed analysis of these different technical approaches (Hartmann & Stead, 2011), building several prototypes and testing performance before finally deciding on a variation of the final hybrid approach that offered a clear split between content and platform. Content can be made by anyone, and is stored either in a generic, mobile format (like EPUB or MP3) or as browser-friendly HTML5. The content is managed by a hand-coded "native" app that provides menus, settings and any other system-level functionality. The open-source PhoneGap framework (www.phonegap.com) is a popular example of this approach. The project team used PhoneGap as the basic building block, adding custom native extensions to add learning-specific functionality. This allowed for a native app, with opening menus hand-coded for each platform but embedded content to be created in a platform-agnostic manner. The bridge between these two layers was managed by PhoneGap and additional plugins developed by the project.

Figure 8.2 shows a snapshot of this in action. The content page shows a video launch page. The black bar on the top is natively coded for iOS. The Android equivalent looks subtly different, with no back button and access to a system menu. The main page itself is created in HTML and is identical on all platforms. When the user clicks the link to watch the video, this can be managed in-page (HTML5) or directly by the app which has performance benefits (launching the video in the native video player).

Figure 8.2: App architecture mapped against a real page from the Global MedAid app.



Open Content: Formats for Mobile Media

The mobile Web and digital media sharing have largely driven the current standards for compressing and sharing mobile media files (W3C standards; see Hazael-Massieux, 2012). Some define the file format itself, and others define how files relate to one another. To maximise the future use of mLearning, these generic mobile formats should be used if at all possible for all new content development.

The basic design principles for looking at content formats were:

- Media should be formatted for cross-platform playback, avoiding platform-specific formats in favour of open ones.
- Individual media files should be optimised for mobile (compressed).
- As much of the interactivity as possible should be delivered via browser-supported technologies. In most cases, this means using HTML5, JavaScript and associated Web formats.
- HTML5 content should be designed to flow, dynamically adapting layout between landscape and portrait, as well as to a range of screen sizes.

Within these broad guiding principles, the project team tested individual media types in an effort to find the optimum format for each one. Video is a good example, since despite standardisation of higher-level formats, not all codecs work on all devices. The team found the optimum formats currently available for each media element to be as follows:

Video and audio – Although there are many different audio and video formats available, most devices (such as the iPod) and programmes (such as Windows Media Player) will only take a few specific formats. An AVI or WMV movie will not play on an iPod, for example, without being converted into an MP4 file. However, there are a few formats that have close-to-universal support from smartphones, and should be used to ensure reusability:

- Audio: The two formats with the broadest support across mobile devices are MP3 (http://en.wikipedia.org/wiki/MP3) and its successor, AAC (http://en.wikipedia.org/wiki/Advanced_Audio_Coding).
- Video: MPEG-4 has close-to-global support on mobile devices, specifically the H.264/AVC standard (http://en.wikipedia.org/wiki/H.264). MPEG-4 itself is a container format, meaning that it can contain many different formats within it, but sticking with H.264/AVC will ensure portability.

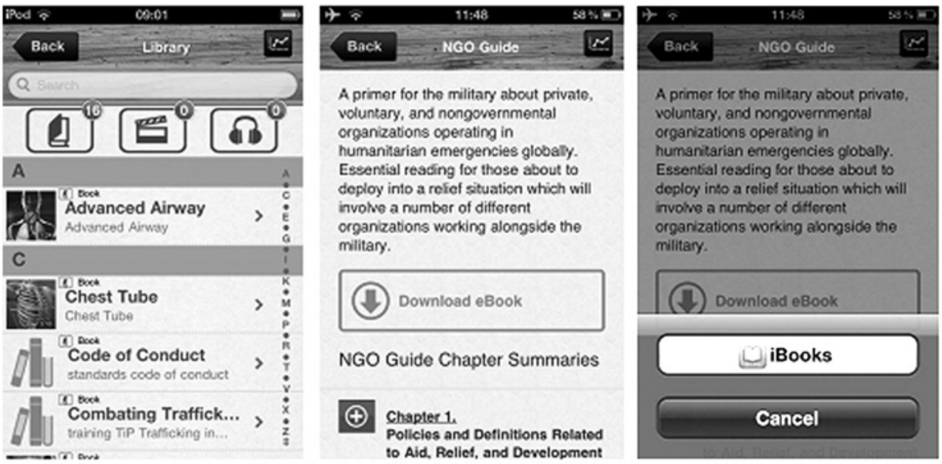
E-books – Where a large amount of text is required, e-books proved an ideal format for sharing and packaging downloadable material. There are many e-book types available, but for maximum coverage the key formats are:

- EPUB: This is rapidly becoming the gold standard. It works on almost all e-book readers (except some Kindles and older phones).
- MobiPocket (.mobi): This is the gold standard for mobile phone based readers. It is not dissimilar to EPUB (some people treat it as an earlier format of EPUB).
- AZW (Amazon's Kindle format): This is exactly the same as .mobi but renamed (can be .mobi, .prc or .azw).

(For more information, see http://en.wikipedia.org/wiki/Comparison_of_e-book_formats.)

All three of these are based on XHTML/CSS (similar to a package of webpages), which means that they may not look the same on all devices. E-book readers themselves are very much like Web browsers but with very limited layout controls. Most only use their own in-built fonts, ignoring other instructions that may be included in the file format. There are several enhanced EPUB formats (like EPUB3 and Apple's new proprietary extensions to EPUB), but these have very little support on smaller devices. See Figure 8.3.

Figure 8.3: Sample workflow integrating a native search, an HTML summary page, and integration with an external e-book reader.



Another option is PDF (portable document format). All modern smartphones can load PDF files directly. This is good (for portability) but very bad for optimised legibility. Unlike the first three formats, the layout is fixed. The page does not reflow to fit the screen. The fonts do not properly resize. This makes for tricky reading.

Feedback from learners reading PDF on their devices was not good. Where possible, it is advised to use EPUB or other flowable formats instead of PDF (EPUBcan be easily converted to .mobi and .azw to reach wider devices).

Open Content: Formats for Embedding Mobile Interactivity

Learning content is rarely just "media." Interactivity is often added to support deeper understanding through the creation of an on-screen activity, or even building a fully functioning learning tool. Use of HTML5 enables all of these options to be supported as both the content and the interactivity travel together.

Another major advantage of the "Hybrid Mobile Web App" approach taken in the project is that, by separating content from platform and allowing the content to be rendered (displayed) in an embedded Web browser inside the app, the content itself can be developed to run equally well on any supported mobile device. It also allows the creation of a rich ecosystem of content. Anything that will work as a Web app can work as mLearning content.

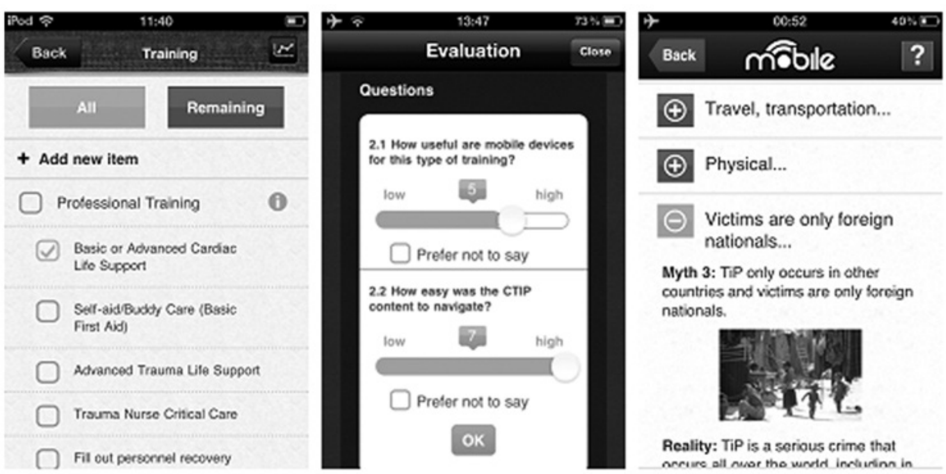
HTML5 as an ideal format – The project defined Mobile Learning Objects (or micro-courses) as self-contained HTML packages not unlike SCORM (an eLearning standard, www.adlnet.gov/capabilities/scorm) or the W3C Widget definition (www.w3.org/TR/widgets). Functionally, this approach is open-ended, allowing developers the freedom to use XHTML, HTML5, and any combinations of CSS and JavaScript to support their content and add richer functionalities.

Any functionality supported by the local Web view (Web browser) is available to course developers. Two different technical approaches are available (see Figure 8.4):

1. Pure HTML, generic JavaScript: By using only HTML and JavaScript with 100% browser support, you can ensure that your content is truly "develop once, play on all devices," but you are limited in the richness of the interactivity.
2. Optimised for different devices: To exploit a wider range of device-specific features, adaptive JavaScript calls can be created that detect the browser type and render optimised pages for each.

Good examples using the second option can be built using JQueryMobile (http://jquerymobile.com) or Sencha Touch (www.sencha.com/products/touch), or perhaps by using WebKit-specific JavaScript calls (www.webkit.org) to achieve animated effects. If developers use these approaches, they can develop richer interactivities, but they need added skills to ensure proper playback across all devices and graceful degradation where these features are not supported.

Figure 8.4: Sample content screens developed using HTML and JavaScript.



These are some of the technological approaches, but to create truly engaging mobile content, significant effort also goes into design and interactivity. Many of the guidelines for making good mobile websites are useful here, though not all the advice is relevant for a downloaded mLearning package.

Useful reference sites for design include:

- Jakob Nielsen's advice on "designing for mobile: www.useit.com/alertbox/mobile-vs-full-sites.html
- Design advice from These Days Labs: http://labs.thesedays.com/blog/2010/07/16/10-tips-for-designing-mobile-websites

Some of the key style guides to consider when using this approach are:

- cut features, to eliminate things that are not core to the mobile use case (requires learning design skills);
- cut content, to reduce word count and defer secondary information to secondary pages (requires editorial skills);
- design with a fluid layout to cope with different screen sizes (minimum width: 320px);
- use of CSS3 for visual effects (rather than older Web-based approaches, like image slices); and
- enlarged interface elements to accommodate "fat fingers" (suggested: 44x44px).

Because the content is displayed via the local browser, developers can test their content by running it live in a browser or by downloading it direct to a mobile device.

Open Sharing: Formats for Packaging and Tracking

ZIP + XML to package mLearning files – For packaging a collection of media and HTML files, it was appropriate to leverage the more established standards for sharing eLearning content (SCORM Content Packaging), which in most cases is done by zipping up a collection of HTML pages and including core metadata to define the content. Some aspects of this approach are perfect for mobile: for example, a single file representing a package of content, in an open, Web-accessible format. Other aspects are not: for example, bloated file formats, excessive metadata, reliance on a SCORM player to support all API calls.

For content packaging, the project used a reduced version of SCORM CP, with a much lighter set of metadata. This allows the content to be entirely stand-alone, in that it can be unzipped to play directly in any mobile browser. But it can also be downloaded and unpackaged by our app, in which case it integrates seamlessly into the learning app, allowing for tracking and monitoring of progress.

Specific data about the XML format is available on the OMLET documentation site (http://omlet.m-learning.net/docs).

Formats for Messaging and Tracking

Traditional eLearning uses the SCORM API as a structured method to pass tracking data from the content to the learning platform. Although widely supported on the big screen, SCORM is not yet widely established in more dynamic learning environments (virtual reality, social media, etc.), or on the smaller screens required for mLearning, and is widely considered too restrictive for tracking the wide range of learning activities typical on a phone (Degani, Martin, Stead, & Wade, 2010). Several parallel initiatives are underway, sponsored by the eLearning industry, to explore alternative methods of sending progress data to a learning platform. Key ones are:

- LETSI (http://letsi.org): protocols for passing progress data back to a learning platform without requiring the content to be hosted by it
- Tin Can (http://scorm.com/tincan): a proposed replacement for the SCORM API, allowing a wider range of content hosted in multiple places to send

more descriptive update on progress. Like LETSI content does not need to be hosted on the tracking site.

Both of these standards are of interest for mLearning. The project team borrowed from each, but did not fully implement either, as these were not core requirements for the project. RESTful Web Services (http://bit.ly/RESTful) were used to exchange information with the Web server (similar to LETSI) and a linear stream of progress updates via a JavaScript API to pass data from the content to the app (like Tin Can).

Combining These Approaches for an Optimum Mobile Learning Format

The combination of technologies and approaches listed above proved to be a fit-for-purpose and effective solution for developing and sharing mLearning content for the target groups (work-based learners using their own smartphones). By leveraging and extending existing standards, open source projects, and appropriate concepts from eLearning, the project team were able to create a new, robust framework for mLearning development, optimised for touch-sensitive smartphones (primarily Android and iOS, but also Windows Phone).

All of the core software developed has been made available as an open-source framework (called OMLET) to encourage future projects to build on the lessons learned and extend them. This includes the back end (online catalogue), the apps themselves (iOS and Android), and sample content implementations. More details are available at http://omlet.m-learning.net/docs.

Conclusion

As mLearning is adopted more widely and the quality improves, it is increasingly important to ensure that good mobile content is transferrable and can work on many devices, across many networks, and in multiple languages. The MoLE project has been working towards the establishment of an open set of standards for mLearning content to allow maximum portability and re-use, without locking out the key features of the most useful tool of all: the phones themselves!

Drawing on existing standards in related domains (mobile Web, HTML, eLearning, video, zip), it has been possible to define formats for both mLearning content, and applications themselves that support open sharing and the future extensibility of mLearning across multiple devices and platforms. By embracing multiple media formats and a wide range of use case scenarios, the best possible learning content can be made available via whichever channel is available to that learner.

The formats and standards proposed do not restrict. They are built from existing, open standards and are being shared to encourage wider consensus and adoption. It is in the same spirit of openness and sharing that the project team has made available the core software frameworks used in the project (and in other commercial deployments), to demonstrate how this approach to mLearning content packages can be successfully adopted.

Currently, over 300 learners across 24 nations are using content developed to these standards (via four different initiatives), and key stakeholders in the U.S. government eLearning community have adopted this approach (and the apps) as core to all their future mobile courses.

To encourage wider adoption in the mLearning community, all the technical details, as well as the software itself, has been released as an Open Source project (OMLET) and extensively documented. The platform and content standards are continuously evolving. Ongoing dialogue and suggestions for improvements to these techniques are always welcome, and contributions to future software development are welcomed via the OMLET website.

Acknowledgments

The author now works at Qualcomm, but would like to acknowledge technical input by previous colleagues at TribalLabs, including Gustavo Hartmann, Mahdi Barakat, Jon Brasted, Octavio Nunes, Gustavo Pilcher, Chris Whitehead, Ben Smith and others.

This research was made possible by research grants awarded to TribalLabs by the Office of Naval Research Global (ONRG) and funded by the Telemedicine and Technology Research Centre (TATRC) at the U.S. Army Medical Research & Materiel Command (USAMRMC), Fort Detrick, as a contribution to the Coalition Warfare programme (CWP) (grant number: N62909-11-1-7026). The views expressed here are solely those of the author and do not represent or reflect the views of any government agency or organisation that participated in the project.

References

ASTD (2012). *Mobile learning: Delivering learning in a connected world*. ASTD research report. Retrieved November 2012 from www.astd.org/Publications/Research-Reports/2012/2012Global

Attwell, G., & Baumgartl, B. (Eds.) (2009). *Creating learning spaces: Training and professional development for trainers*. Vol. 9. Vienna: Navreme Publications.

Blake, C., Adams, A., & Scanlon, E. (2011). Effectiveness of mobile learning across various settings. In CAL Conference 2011, *Learning Futures: Education, Technology & Sustainability,* 13–15 April 2011, Manchester.

Bradley, C., Haynes, R., Cook, J., Boyle, T., & Smith, C. (2009). Design and development of multimedia learning objects for mobile phones. In M. Ally (Ed.), *Mobile learning: Transforming the delivery of education and training* (pp. 157–182). Edmonton, AB: Athabasca University Press. Retrieved June 11, 2012, from www.aupress.ca/index.php/books/120155

Cook, J. Holley, D., Smith, C., & Bradley, C. (2006). *A blended m-Learning design for supporting teamwork in formal and informal settings*. Paper for IADIS International Conference on m-Learning, 2006.

DeGani, A., Martin, G. Stead, G., & Wade, F. (2010). E-learning standards for an m-Learning World. Archived paper at www.m-learning.org/knowledge-centre/research

Good Technology. (2012). State of BYOD report. Retrieved January 2013 from www1.good.com/resources/whitepapers

Hartmann, G., & Stead, G. (2011). Cross platform mobile app development. Retrieved November 2012 from www.triballabs.net/2011/07/cross-platform-mobile-app-development/

Hazael-Massieux, D. (2012). Standards for Web applications on mobile. Synopsis of W3C standards. Retrieved from www.w3.org/2012/05/mobile-web-app-state

Koole, M. L. (2009). A model for framing mobile learning. In M. Ally (Ed.), *Mobile learning: Transforming the delivery of education and training* (p. 38). Edmonton, AB: Athabasca University Press.

Kukulska-Hulme, A. (2010). Mobile learning as a catalyst for change. *Open Learning 25*(3), November 2010, 181–185.

Pachler, N., Pimmer, C., & Seipold, J. (Eds.). (2011). *Work-based mobile learning: concepts and cases*. Oxford: Peter Lang.

Savill-Smith, C., Attewell, J., & Stead, G. (2006). *Mobile learning in practice: Piloting a mobile learning teachers' toolkit in further education colleges*. London: Learning and Skills Network.

Sharples, M., Taylor, J., & Vavoula, G. (2005). Towards a theory of mobile learning. In H. van der Merwe & T. Brown (Eds.), *Mobile technology: The future of learning in your hands*, mLearn 2005, 4th World Conference on mLearning, 25–28 October 2005. Cape Town: mLearn.

Smith, C., Bradley, C., Cook, J., & Pratt-Adams, S. (2011). Designing for active learning: Putting learning into context with mobile devices. In A. D. Olofsson & J. O. Lindberg (Eds.), *Informed design of educational technologies in higher education: Enhanced learning and teaching* (pp. 307–329). Hershey, PA: IGI Global.

Stead, G. (2012a). Mobile learning: Mind the gap. Retrieved November 2012 from http://bit.ly/mobimoocstead

Stead, G. (2012b). Open standards for m-learning? Retrieved November 2012 from http://bit.ly/openmob

Towards Maturity (2013). Mobile learning at work. Retrieved January 2013 from www.towardsmaturity.org/article/2013/01/15/towards-maturity-focus-report-mobile-learning-work/

Traxler, J. (2011). The "learner experience" of mobiles, mobility and connectedness. Retrieved November 2012 from http://mobimooc.wikispaces.com/MobiMOOC+2011+(archived)