Towards open formats for Mobile Learning

Geoff Stead
Tribal Group
Cambridge, UK
geoff.stead@tribalgroup.com

ABSTRACT

Much of the current discourse in the emerging field of 'mobile learning' looks at traditional learning, happening in existing educational institutions. A parallel (and less well understood) tribe of mobile learners are adults at work, using their own, personal technology to access critical information at their moment of need for 'just in time' and 'as and when necessary' training (Wishart and Green, 2009).

As smartphones become more and more ubiquitous, and the boundary between work and leisure becomes more and more blurred, the expectations of devices and what they can do are higher than they have ever been. Users are tending to use their devices for all aspects of their lives: work, personal organisation, leisure activities, communication, recording etc, and make less distinction between these different activities than before. And as Traxler says, "mobile devices demolish the need to tie the particular activities to particular places or particular times" (Traxler 2011).

This blend between "work" and "play", and increased expectations that any learning apps accessed on your private phone should "perform" as well as any other app, places an impossibly high expectation on educators, learning technologists and mobile developers interested in supporting mobile learning. How to craft a mobile app experience that rivals the best commercial apps available, yet offers rich and pedagogically sound access to resources in a cost effective manner? To do this requires a new methodology that allows mlearning content to "travel well" between platforms, and device types, and provides solutions to a broad range of technology challenges, some of which are unique to mlearning.

As part of their work on the US Government funded Mobile Learning Environment (MoLE) project, the author and his technology team have been wrestling with many of these challenges, whilst creating a suite of mobile content, performance support tools, and job aids designed to be used across many nations, languages and devices. Their software is currently in use in over 20 nations, preparing emergency workers for disaster situations.

Their findings, and technical solutions are already the basis of an open framework for defining and building mobile learning content. This paper describes some of the technical challenges and shares a possible foundation for open sharing and reuse of mobile learning content.

Author Keywords

mlearning, apps, work-based learning, open formats, html5, cross platform development, mobile learning objects

INTRODUCTION

The US Government funded a 2-year technology research project to explore the technical challenges involved in transitioning mobile learning from a peripheral, exploratory study into a core part of their mainstream e-learning delivery. This was done via the Mobile Learning Environment project (MoLE, www.mole-project.net) which developed sample content, tools and platforms for work-based learners involved in humanitarian and disaster relief work. These were deployed as Global MedAid, a mobile app for both iOS and Android currently in use across 20 nations, with 600 learners and multiple languages.

As part of the work, the technology team did practical research and prototype development looking at the underpinning technologies required to deliver meaningful mobile learning tools and content across a range of platforms to a massively diverse user group. Work-based learners have very specific requirements of mobile learning, needing small, easy to access, nuggets of learning and support tools that are quick to locate, and easy to use across a wide range of devices. Work-based learning is multi episodic, often informal and takes place on a just-in-time basis. Although mobile devices are known to foster situated approaches to learning in and across work contexts (Pachler et al., 2011), the employers of these particular learners had previously not allowed this. In addition, the identified user group were expecting to have only occasional access to the internet, and needed to use a mix of resources including:

- compliance-type "courses" that require tracking
- video interviews with domain experts
- active "checklists" as performance support tools
- eBooks, and other existing resources
- mobile reference tools and lookup charts.

Previously, our target group had no mobile access any of these resources. Whilst the mobility of the potential users presents some challenges it also provides opportunities to enhance performance, and contextualise learning. As Kukulska-Hulme points out, the notion of mobility relates not just to physical mobility (of the device or the user), but the opportunity to overcome physical constraints by having access to people and digital learning resources, regardless of place and time (Kukulska-Hulme, 2010).

In designing and developing the optimal set of tools and content for these users, the research team was able to propose frameworks and standards that would apply to a wider range of work-based mobile learners. These standards, though still evolving, have already been adopted by the leading US military e-learning platform as part of a move to mobile. This paper sets out the basic foundations, shares technical lessons learned and outlines the proposed open formats.

TECHNOLOGY AS AN INTEGRAL DIMENSION

When evaluating emerging technologies as tools for learning, it is impossible to separate out the learning from the technologies. This is true from an academic perspective, as well as a technical perspective: the effectiveness of the full learning experience is a complex blend of the learner's own skills, the affordances of the device, the appropriateness of the content, the context of the learning, the fluidity of the software, and the performance of the mobile app itself. Some frameworks for mobile learning (like FRAME by Koole, 2009) make reference to this inter-relationship, showing how the mobile learning is an interaction between the technology, the learner and the context.

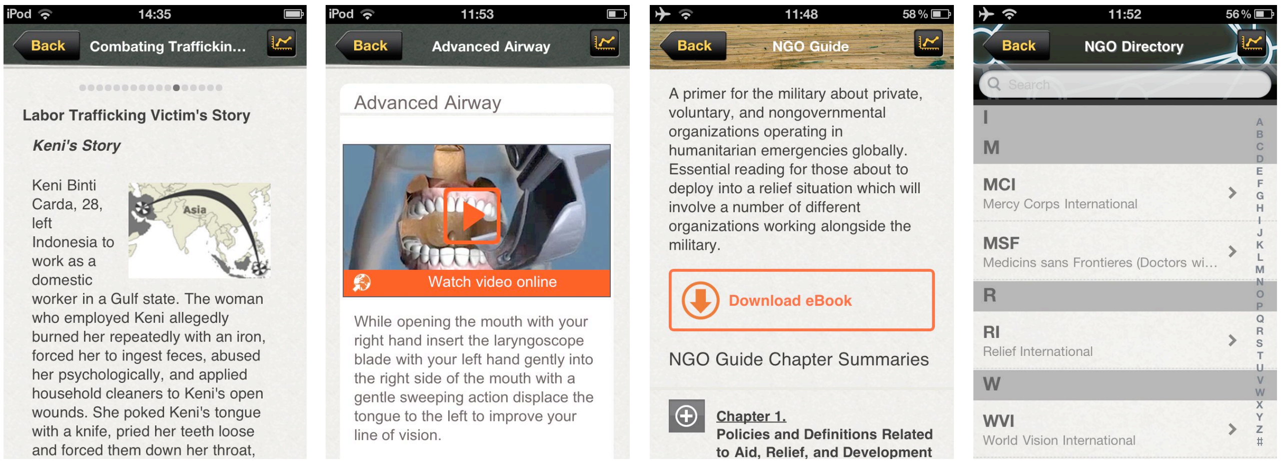
This paper emphasises the links between these, with a firm focus on the underpinning technologies, and their appropriateness for mlearning. We look at content issues (like data formats), as well as the technologies required to create interactivities around the content. We look at user interface design, and how this differs across mobile platforms. We look at protocols for sharing packages of mobile content between phones, as well as mechanisms to share tracking data with learning platforms.

When evaluating mobile learning content, we found it beneficial to encompass as many of these dimensions as possible. And when extrapolating that into a strategy for mobile content, some of the more abstract technical dimensions should be included, as well as the learning dimensions. Examples of this would be: ease of navigation; quality of interactivities, and appropriateness for that specific device; "findability" of content on a specific device; range of supported devices; effort required to move content onto a new mobile device. This provides challenges for all the stakeholders in any mobile learning technology development. Some typical dilemmas include:

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

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

Unlike the more traditional e-learning model, mobile learning does not expect one single app, or course to provide the entire breadth of learning content. It is assumed that a small amount of well-contextualised information is better for learning than a larger, and more general course, and that part of the benefit of mobile access to info is being able to jump into a small nugget of personalised information, rather than a larger, more structured course. (see Sharples, Taylor & Vavoula, 2005 for a typical example). This makes it all the more relevant to support a larger pool of small learning nuggets that can be seamlessly assembled in different combinations, for different learners, on different devices.


iPod 14:35
Back Combating Traffickin...
Labor Trafficking Victim's Story
Keni's Story
Keni Binti Carda, 28, left Indonesia to work as a domestic worker in a Gulf state. The woman who employed Keni allegedly burned her repeatedly with an iron, forced her to ingest feces, abused her psychologically, and applied household cleaners to Keni's open wounds. She poked Keni's tongue with a knife, pried her teeth loose and forced them down her throat,

iPod 11:53
Back Advanced Airway
Advanced Airway
Watch video online
While opening the mouth with your right hand insert the laryngoscope blade with your left hand gently into the right side of the mouth with a gentle sweeping action displace the tongue to the left to improve your line of vision.

11:48 58 %
Back NGO Guide
A primer for the military about private, voluntary, and nongovernmental organizations operating in humanitarian emergencies globally. Essential reading for those about to deploy into a relief situation which will involve a number of different organizations working alongside the military.
Download eBook
NGO Guide Chapter Summaries
Chapter 1.
Policies and Definitions Related to Aid, Relief, and Development

11:52 56 %
Back NGO Directory
Search
I
M
MCI
Mercy Corps International
MSF
Medicins sans Frontieres (Doctors wi...
R
RI
Relief International
W
WVI
World Vision International

Figure 1: Range of different types of content from Global MedAid app

TECHNICAL APPROACHES TO SHARING MOBILE LEARNING

For learning technologists developing mobile learning apps, there are three main approaches to technical development:

1) Open apps: software techniques enabling developers to build an app once, that is able to run on different mobile phone platforms, (cross-platform development).
2) Open content: content formats that allow individual pieces of content to display on multiple devices. Either via industry standard "players" (like eBook readers), or even better, with native device support (like audio files)
3) Open content with embedded interactivity: this is often seen as a hybrid between the above two approaches, and is the ideal scenario for learning interactivities, as it allows a combination of both content and appropriate learning interactions.

All three of these were explored during the technical developments of the MoLE project, and the core findings follow.

OPEN APPS: APPROACHES TO CROSS PLATFORM APPLICATION DEVELOPMENT

Despite many technology enthusiasts engaging in this area, there remains no perfect answer to the question of how to develop a mobile application once that will work on all phones. Despite significant work from players including the W3C Core Mobile group (http://www.w3.org/community/coremob) and the Open Mobile Alliance (http://www.openmobilealliance.org) to promote best practices and mobile standards, there remains no simple solution to cross platform development, and no consistent guidelines or frameworks to address common problems like the delivery of cross-platform content that works seamlessly on any device.

As each vendor implements its own application development stack, achieving cross-platform and cross-device consistency is a non-trivial task. Fortunately as the web becomes ubiquitous and its technologies evolve, with more and more mobile browsers implementing new standards like HTML 5, CSS 3 and JavaScript, web applications are rapidly becoming an attractive and cost-efficient way of developing mobile applications, that can rival native apps in terms of rich user experience and access to advanced capabilities like storage and geo-location. Fortunately, mobile learning enthusiasts are not the only people looking at this challenge, and a range of tools and frameworks to support cross-platform development have flourished and evolved over the last couple of years. The main approaches that are currently available are outlined below (Hartmann &Stead, 2011).

Cross Compilation (code once, generate native apps)

A cross-compiler separates the build environment from the target environment, effectively decoupling a source from its target. The mobile app developer codes in a third language (like JavaScript, Ruby or Java), using a special API (Application Programming Interface) to build the mobile application, including the user interface, data persistence and business logic. The code is then processed by a cross-compiler that transforms it into platform-specific native apps for the different platforms that the application will run on. The software artefact generated from this process can be deployed and executed natively on the device.

The advantages of this technique are: performance, as the application is running natively on the device; improved user experience, since the app behaves like a regular app on the user's ecosystem; and full native access to a range of device specific capabilities like integrated camera, sensors, etc. The big disadvantage is complexity, since cross-compilers can be difficult to write and need to be kept consistent with the fragmented mobile platforms and operating systems available.

High profile platforms offering this approach include: Appcelerator Titanium; Rhodes (rhomobile)

Mobile Web Apps (run in the mobile browser)
Another increasingly popular approach is to build the app as a mobile web application that will run on the user's mobile browser. This involves using standard web technologies like HTML, CSS and JavaScript to build the application and make it look and behave like a native app. This is possible due to the advanced capabilities of HTML 5 and CSS 3, including embedded SQL databases, local storage, animations, canvas, web sockets and video playback. Although HTML 5 is still a young technology (the standard is yet to be finalised) and mobile browsers may implement it differently, its increasing popularity in rendering engines like the WebKit (which powers the iPhone and Android mobile browsers) allow web apps to look and behave more and more like native apps.

For certain classes of application this approach is appealing, as it is quick to develop and potentially covers a wide range of platforms with minimal changes. This includes common business applications like news readers, e-books, mobile banking, social interaction and e-mail. However it is less suited for highly interactive, CPU-intensive and visually rich applications like games, augmented reality browsers and videoconferencing.

Web apps would typically run in a standalone mobile web browser (pure web). The web approach brings some advantages, like simplified deployment and immediate availability, since most modern phones come with a browser installed and to run the app the user just needs the URL and an active data connection to get started. The big drawback would be a poorer user experience (the browser is never as interactive as the native phone), and restricted access to advanced device capabilities like contacts, storage and sensors.

Hybrid Mobile Web Apps (Web-style content embedded in a native app)
A variation of the above is to have a native app that embeds a browser inside it, allowing for some of the advantages of a natively built app, together with the benefits of portability that come with web-based content (hybrid web). In this hybrid model, the web app runs inside a thin "wrapper style" native app which provides a bridge to the device's operational system and services. The web application is cached locally on the device on installation, removing the need for an active data connection and improving its speed and responsiveness. The communication between the web app and the native app normally happens over JavaScript via custom built APIs.

This technique combines the best of both worlds into one single integrated solution: flexibility of web apps with speed and feature richness of native apps. This approach can offer wider support over many devices, without needing to redevelop the content itself. It also allows developers to compensate for failings in mobile browsers on specific devices by adding extra "native" features where the mobile browser cannot cope.

A high profile framework offering this approach is PhoneGap (http://phonegap.com/), recently acquired by Adobe.

Mobile Widgets
Finally, leveraging web technologies, vendors have created another way for mobile web sites to run like native installed applications. The approach has many different names, but is normally referred to as 'mobile widgets'. The widget concept was introduced long before the mobile app and app store revolution and can be seen as a first stab at delivering small nuggets of functionality in a lightweight and intuitive way to the end user. Popular mobile widget platforms include: Symbian/Nokia Web RunTime engine, Sony Ericsson's Xperia widget engine, BlackBerry's Widget SDK and Samsung's TouchWiz widgets.

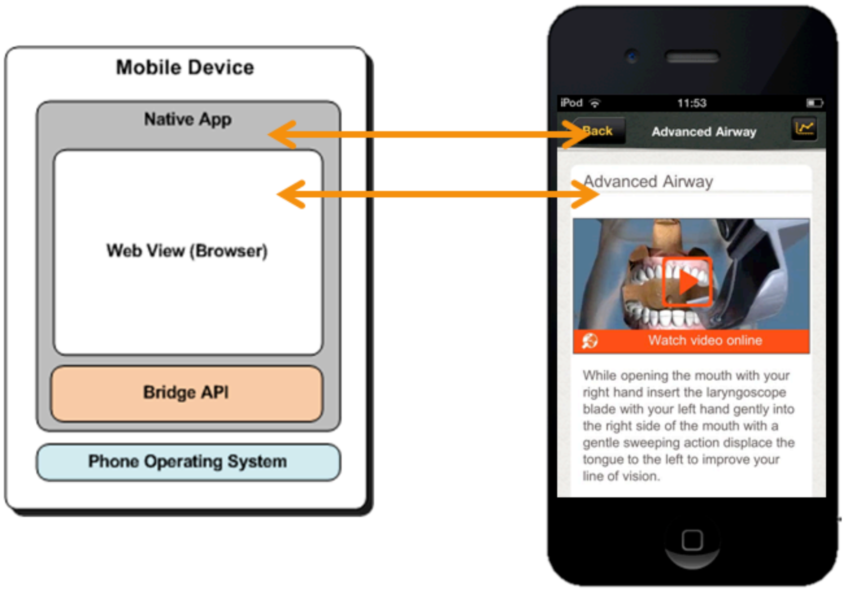
A widget is an interactive tool that provides a single-purpose service to the user, such as showing the latest news, current weather, date and time, calendar, dictionary, map, calculator or even a language translator [Wikipedia]. On mobile phones, widgets normally appear on the home screen or virtual desktop. Mobile widgets are small apps normally written using standard HTML, JavaScript and CSS. The use of web technologies is invisible to the user, and the application can work just like any other software installed on the device. The platforms normally provide JavaScript APIs, so widgets can access device capabilities such as the camera, contacts and storage, like a regular native app. This is very similar to the hybrid approach described earlier, the only difference being the packaging and access to phone capabilities, since normally widget API's are richer than bespoke cross-platform libraries based on HTML 5.

Although standards were created to help promote and standardise the widget landscape, they are still not widely adopted by vendors, creating a similar fragmentation to that in the mobile app development world. Fortunately, since most widget engines leverage JavaScript, it is possible to reuse almost all the code, creating a multi-platform widget engine.

Summary
The project team did a detailed analysis of these different approaches, building several prototypes, and testing performance, and ended up selecting a blend of several approaches:

For our own apps, we now use the hybrid model, keeping all mobile content in HTML format, but hand-coding 'native' app code to provide any system level functionality. We use the open source PhoneGap framework as the basic building block, but then create custom native extensions to add learning-specific functionality.

This allows us the performance benefit of native app features, without sacrificing content portability. For our mobile content/learning objects, we used the formats defined in Web Apps, or Widgets allowing the same content to play in multiple apps, on many different mobile platforms.



While opening the mouth with your right hand insert the laryngoscope blade with your left hand gently into the right side of the mouth with a gentle sweeping action displace the tongue to the left to improve your line of vision.

Figure 2: App architecture mapped against a real page from Global MedAid app

OPEN CONTENT: FORMATS FOR MOBILE LEARNING CONTENT

There are many established and emerging standards for compressing and sharing mobile media files. Some define the file format itself, and others how files relate to one another. To maximise the future use of mobile learning, these formats ought to underpin any new mobile learning development. We thus developed "mobile content guidelines" that were shared by all project partners, and ensured the base format of all mobile learning content would be governed by 'mobile web standards', regardless of whether the content itself is destined for the web. The basic design principles selected were:

- Individual media files should be optimised for mobile, and stored in as open a format as possible.
- Media should be formatted for cross-platform playback, avoiding platform-specific formats.
- 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.
- Content layout should flow (dynamically adapting between landscape and portrait, as well as a range of screen sizes)

For specific media types, we tested multiple formats across a range of platforms, and found the optimum formats to be:

Video and audio

There are many different audio and video formats available, and most devices (such as the iPod) and programs (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. There are a few formats, however, that DO have close-to-universal support from smartphones.

Audio	File format: MP3 or AAC	http://en.wikipedia.org/wiki/MP3 http://en.wikipedia.org/wiki/Advanced_Audio_Coding
Video	Video Codec: H.264 File format: MP4; Audio codec: AAC	http://en.wikipedia.org/wiki/H.264

eBooks
Where a large amount of text is required, eBooks proved an ideal format for sharing and packaging downloadable, text-based materials in a real, global format. There are multiple formats available for eBooks, but for maximum coverage these are the key formats (http://en.wikipedia.org/wiki/Comparison_of_e-book_formats):

- ePub: rapidly becoming the gold standard. Works on almost all eBook readers (except some Kindles, and older phones):
- MobiPocket (.mobi): The gold standard for mobile phone based readers. It is not dissimilar to ePub (some 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)

All three of these are based on XHTML / CSS (so a bit like a package of web pages). Although the eBook readers themselves are very much like web browsers, they have limited layout controls. Most only use their own inbuilt fonts (ignoring what you tell them). Another option is:

- pdf (Portable Document Format): All modern smartphones can load pdf files directly. This is good (for portability), but very bad for mobile optimized legibility. Unlike the formats above, the layout is fixed. The page does not reflow to fit the screen. The fonts don't really resize meaningfully. This makes for tricky reading

We found the best format was ePub (though we would also convert the same file to .mobi and .azw to reach wider devices).

OPEN CONTENT WITH EMBEDDED INTERACTIVITY: FORMATS FOR MOBILE LEARNING CONTENT

One major advantage of the "Hybrid Mobile Web Apps" approach mentioned above is that the mobile content itself can be rendered (displayed) in an embedded web browser inside the app. This allows for a rich ecosystem of content. Anything that will work as a web app can work as mobile learning content.

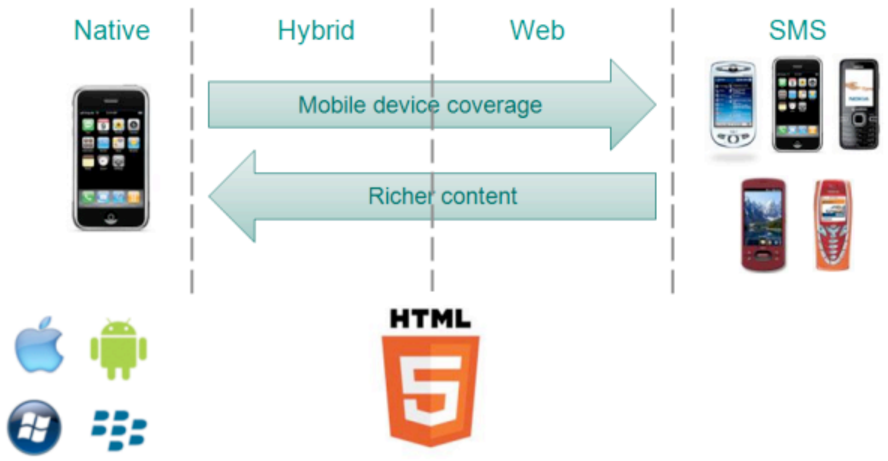


Figure 3: Spectrum of mobile technologies against reach

Figure 3 shows a typical technology spread. On the left, a hand-coded native app offers the richest functionality, but for one type of device only. On the far right, a simple text message can reach any mobile device, but with very limited ability for interactivity. In the middle are the web technologies, either delivered online or via a hybrid app.

HTML5 as an ideal format:
Because of this, we defined Mobile Learning Objects (or micro-courses) as self-contained HTML packages (not unlike SCORM packages (an e-learning standard), or the W3C Widget definition). This is extremely open ended, allowing developers the freedom to use XHTML, HTML5, and any combinations of CSS and JS to support their content, and add richer functionalities.

Any functionality supported by the local webview (web browser) is available to course developers. There are 2 different technical approaches that can be adopted:

1. Pure HTML, generic JS: by only using 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 JS calls can be created that detect the browser type, and render optimised pages for each.

Good examples of this are by using JQueryMobile (http://jquerymobile.com), Sencha Touch (http://www.sencha.com/products/touch) or perhaps using WebKit specific JS calls (http://www.webkit.org) to achieve animated effects. If developers use these approaches they can develop richer interactivities, but need added skills to ensure proper playback across all devices and graceful degradation where these features are not supported. These are the technological approaches, but to create truly engaging mobile content, a lot of effort needs to go into the design and interactivity. Many of the guidelines for making good mobile websites are useful here, though not all advice is relevant for a downloaded mobile learning package.

Useful reference sites include:

- Jacob Nielsen's advice on 'designing for mobile': http://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 main points to consider are:

- cut features, to eliminate things that are not core to the mobile use case;
- cut content, to reduce word count and defer secondary information to secondary pages;
- design with a fluid layout to cope with different screen sizes (min-width: 320px);
- use CSS3 for visual effects (rather than older web based approaches, like image slices);
- enlarge interface elements, to accommodate 'fat fingers' (suggest 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.

ZIP + XML to package mlearning files:

For packaging a collection of media and HTML files, we turned to the more established standards for sharing e-learning 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 (a single file representing a package of content, in an open, web accessible format). Other aspects are not (bloated file formats, excessive metadata, reliance on a SCORM player to support all API calls).

For our content packaging, we used a reduced version of SCORM CP, with a much lighter set of metadata. This allows the content to be entirely standalone, 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.

Formats for messaging and tracking:

Traditionally e-learning used 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 for traditional e-learning, SCORM is not yet widely established in more dynamic learning environments (virtual reality, social media, etc), nor on the smaller screens with mobile learning, and is widely considered too restrictive for tracking the wide range of learning activities typical on a phone (Degani et al, 2010). Several parallel initiatives are underway, sponsored by the e-learning 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 replacement for the SCORM API, allowing a wider range of content 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 mobile learning. We borrowed from each, though did not do a full implementation of either, as these were not core requirements for the project. We used Restful Web Services to exchange information with our 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

For our target groups (work-based learners using their own phones) we found the combination of technologies listed above, a perfect solution to developing and sharing mobile learning content. We leveraged and extended existing standards, and open source projects where available, and borrowed concepts from e-learning where helpful. Our focus was specifically on touch-sensitive smartphones, (primarily Android and iOS, but also Windows Phone).

CONCLUSION

As mobile learning is adopted more widely, and the quality of mobile learning content is enhanced, it is becoming increasingly important to ensure that good quality mlearning content can work on many devices, across many networks, and in multiple languages. The MoLE project has been working to establish an open set of standards for mobile learning content to allow for maximum portability and reuse, without locking out the most useful aspects of mlearning: the phone features themselves!

Drawing on existing standards in related domains (mobile web, html, e-learning, video, zip) it has been possible to define formats for both mlearning content, and applications themselves that allow for open sharing and 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.

At the time of writing, over 300 learners across 24 nations are already using content developed to these standards (via four different initiatives), and key stakeholders in the USA government e-learning community have adopted this approach (and our app) as core to all their future mobile courses.

The platform and content standards are still evolving, and an ongoing dialog and improvements to these techniques is always welcomed.

ACKNOWLEDGMENTS

The author would like to acknowledge technical input by Gustavo Hartmann, Mahdi Barakat, Jon Brasted, Octavio Nunes, Gustavo Pilcher, Chris Whitehead, Ben Smith and others in the TribalLabs team.

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

REFERENCES

Attwell, G. & Baumgartl, B. (Eds.) (2009): Creating Learning Spaces: Training and Professional Development for Trainers. Vol.9, Navreme Publications, Vienna

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

Cook, J. Holley, D., Smith, C., Bradley, C. (2006). A Blended m-Learning Design for Supporting Teamwork in Formal and Informal Settings (Conference paper for IADIS International Conference on m-Learning, 2006)

Hartmann, G. and Stead, G. (2011). Cross Platform Mobile App Development. Archived report on TribalLabs site http://www.triballabs.net/2011/07/cross-platform-mobile-app-development/

Koole, M.L. (2009). A Model for Framing Mobile Learning, in Ally, M. (ed.), *Mobile Learning: Transforming the Delivery of Education and Training*, Edmonton, 2009, p.38

Kukulska-Hulme, A. (2010). Mobile learning as a catalyst for change. Open Learning. Vol. 25, No. 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 and Stead, G. (2006) Mobile Learning in Practice: Piloting a mobile learning teachers' toolkit in further education colleges.

Sharples, M., Taylor, J., & Vavoula, G. (2005) Towards a Theory of Mobile Learning. In H. van der Merwe & T. Brown, *Mobile Technology: The Future of Learning in Your Hands*, mLearn 2005, 4th World Conference on mLearning, Cape Town, 25-28 October 2005. Cape Town: mLearn.

Traxler, J. (2011). The 'Learner Experience' of Mobiles, Mobility and Connectedness. Archived article on MobiMooc site http://mobimooc.wikispaces.com/MobiMOOC+2011+(archived)

Wishart, J. and Green D. (2009). Emerging Issues in Mobile Learning: Future Scenarios for Work Based Learning. In N. Pachler and J Siepold Eds., *Mobile learning cultures across education, work and leisure*, Book of abstracts, 3rd WLE Mobile Learning Symposium, London, 27th March 2009.