The 8th International Scientific Conference
eLearning and software for Education
Bucharest, April 26-27, 2012
10.5682/2066-026X-12-104

MLMS – ENABLING SCORM COMPATIBLE LEARNING MANAGEMENT SYSTEMS

Dragoş CAZACU, Anatol OPREA, Aurelian JURCOANE
ASCENDIA Design, 34 I.C. Brătianu Av., Bucharest, Romania
E-mail: dragos.cazacu@ascendia.ro, anatol.oprea@ascendia.ro, aurelian.jurcoane@ascendia.ro

***Abstract**: Starting with 'iFever', during the last five years, mobile devices became the next big step in the ITC industry with the next five anticipating their becoming the "de facto" Internet access devices. eLearning must migrate as well, in order to maintain its main purpose: delivering information and empowering knowledge acquisition. The mobile enabled Learning Management Systems must be a hybrid one, capable of delivering content and record user progress in various contexts both classical and mobile. Although the main focus in present times has been on how content should be designed for mobile delivery, the same question should be asked for Learning Management Systems, as well, how they should be designed to enable delivery of such mobile content. This paper analyses what are the main challenges for designing such an mLMS are, and how they can obey both the quite old requirements like the SCORM standard as well as the mobile devices characteristics such as discontinuous access to the LMS (due to Internet access limitations) within a session or between sessions, content faceting (adding mobile support for existing content), feedback from and synchronisation of multiple access points (mobile or desktop) in achieving learning goals and objectives by using cloud enabled architectures. We are addressing issues concerning content compatibility validation for mobile devices, aggregation of mobile content with analogous content, enhancement of existing content and how an mLMS should be storing asynchronous submission of learner progress data as well as synchronicity between desktop clients and mobile clients. The last concern involves analysing device delivery mechanisms, custom LMS clients for mobiles and capabilities that could be implemented by such clients in order to overcome the inherent limitations of a browser-only web application delivery system for mobile devices. Last but not least we address issues concerning accessibility of content, a feature often overlooked by mobile content in general.*

***Keywords:** SCORM, mobile, LMS, cloud, eLearning*

I. INTRODUCTION

Ever since the inception of eLearning, the client-server architecture - where the server does the learning management, and the client delivers the content to the user based on the server's decisions - is the most accepted solution. LMS, LCMS, LAMS, VLE, or any other acronym used to describe such a system, use an application server and a set of resources based on pedagogical principles delivered as courses, linked by a communication protocol (standard or not).

The latest smart devices (smart phones, tablet computers or other Mobile Internet Devices) have led to a new branch: mLearning. Without being clearly defined and without a set of principles, mLearning represents today all the technologies and educational applications designed for "smart-tablets": stand-alone applications, video streaming clients, PDFs, different university course formats, mobile application for classic LMS or interactive e-books.

This current situation is similar to what had occurred in the 90's, when all types of digital distribution were considered eLearning. 20 years later, the new technologies require a new order and classification. Information is now more than eLearning. There is a new and undefined area.

The purpose of this paper is to present the challenges of LMS developers in order to adapt to mobile devices. Developers have to switch from a compatible SCORM eLearning solution to a hybrid one: a hybrid between "classic" eLearning and the new up-comings: smartphones and computer tablets.

II. A BRIEF OVERVIEW OF THE ELEARNING DURING LAST FIVE YEARS

In 2007, eLearning reached its maximum. After about 15 years of slow evolution, eLearning was finally widely accepted. Under economic growth, public and private institutions allocated impressive budgets to research, development and eLearning implementation. SCORM was used in its two major versions on 90% of the global platforms [1]. The changing rate from SCORM 1.2 to the much more advanced SCORM 2004 (already at its 3rd Edition) was promising enough to make platform, content or other tools developers innovate continuously in their products. They integrated social media and considered new technologies to improve users' experience.

In 2008 and 2009, the economic crisis affected the eLearning environment: with severe budget cuts, the developers focused on new solutions capable to increase productivity despite the massive personal loss.

In 2010 and 2011 the iPad established a new standard for smart mobile devices: the focus was on how to innovate in the field of mobile platforms, the only field with exponential growth [2]. The eLearning developers were confronted with two major problems: the severe budget cut and the radical technological change. They found themselves exceed by these problems. Most of their efforts were to improve the existing solutions, in order to assure a minim of compatibility with the new technologies. They also wanted to assure a diminished access from the mobile terminals to a subset of LMS functionalities.

Today in 2012 AD, the initiative belongs to the environment's owners: Apple, with the 'iTunes U' program and textbooks, has now the pole-position in the mLearning paradigm [3].

III. THE NEW WONDER CHILD: MLEARNING

mLearning is a new paradigm who is yet to be defined. It includes any ecosystem which enables learning content delivery to the learners via mobile devices, from direct content distribution in mobile application stores, video tutorials using mobile ready video formats, interactive books using proprietary formats to LMS based solution with interfaces adapted to run on mobile devices' screens.

But there isn't a common set of rules; each company has its own standards, formats and specifications. The best example is Apple, with its learning ecosystem based on iPad, iTunes U, iBook Author and textbooks. The company has created a distribution channel and a series of free development tools that allow the content developers and the universities to connect with the final users. This is a huge step for mLearning that could also leave unemployed half of the eLearning developers. Assuming there is no concern about the iPad only solution availability, could this ecosystem be the best solution for every user?

Yes, if we were still in the 90's. This solution has the delivery channel, the potential for highly immersive content, the Learning System but it hasn't got a very important thing: the management. It is a one-way communication highway. The learning objectives are not monitored [4]. Or, if they are, the monitoring process is slow and appropriate only for small groups. So, this is the perfect solution for edutainment and for those "good learners" who can learn without being monitored. It is not a solution for large, heterogeneous groups where the need of curricula objective accomplishment is a must, such as academic or corporatist environments. In these environments the SCORM eLearning solutions have a huge rate of success.

This is why mLearning is the wonder child of today's technology. It all depends on how its potential is used. Let us use it correctly, for it to grow and develop properly.

IV. TO CHANGE OR NOT TO CHANGE

Existing eLearning developers targeting mobile delivery have identified two strategies for enabling such a delivery to take place:

The path of least resistance - Delivering content by traditional means using mobile phone browsers that abide to HTML and JavaScript standards, thus respecting current restrictions resulting from lack of native hardware capabilities (sensors, GPS, camera, gyro, etc.) as well as lack of native operating system APIs (social media integration, push, notifications and reminders, etc.). In this approach all existing Learning Management Systems even those not targeting mobile device delivery, have a chance to perform such a delivery.

Delivering content this way does not provide a good experience for mobile users, sometimes existing content may not run well if at all, especially when taking into account Flash content (present in almost 90% of all interactive content delivered by an LMS) which cannot run officially on iOS browsers. By the other hand, there are reported high level of satisfactions from users attending a mobile enabled LMS course, in terms of mobility and overall user experience [5].

The path of many paths (Zen pun intended) - Developing separate solutions for each mobile environment, solutions that are capable of exchanging and centralizing information by means of platform independent protocols or platform accepted protocols. This direction, while taking full advantage of native hardware and OS capabilities, presents great challenges to content developers who find it hard to provide a streamlined experience across all platforms (since not all platforms provide the same functionalities) and must spend additional time that is to some extent proportional with the number of platforms designated for deployment (the same issue present with native mobile games).

All these differences lead to different publishing timelines and different versions, meaning some users and platforms may even get ignored or lag behind, making the chosen solution purely market driven and without any real value. Thus the need for a unified solution integrating the two approaches above into an ideal mLMS design.

Before tackling the real issue of an ideal mLMS design, we must ask ourselves the question: Is the client-server model still feasible? Given different environments for delivery with different operating systems and different content authoring technologies and frameworks, is there a good approach that can provide platform independence for delivery and provide early adoption of emerging technologies such as 3D graphics, cloud based computing, new paradigms and devices which are taking a steady but growing share from the ITC market and implicitly taking away from the old desktop computer experience?

The answer is a resounding “yes”! The web is ever changing, with new internet enabled devices coming out all too often, new distribution channels being opened all the time and new technologies that improve the interaction between users and content being introduced every month, and yet HTML + JavaScript solutions are still on the edge of the wave being quint essential to everything “web” and even managing to provide the preferred Flash alternative in the mobile world (Adobe chose to focus on desktop player and to provide native compiler options for major mobile platforms) [6]. That is why eLearning in general and SCORM in particular, which rely on these standards still can benefit of this visionary choice; this is what we identify as an important factor to lay at the foundation of mLMS and eventually mLearning successful designs.

V. THE MOBILE ENABLED LMS

The mobile enabled LMS must be a hybrid capable of delivering content and track user progress in various contexts (desktop, mobile). Although the main focus in present times is on how the

content should be designed for mobile delivery, the same question should be posed for a Learning Management System; how the system should be designed to enable delivery of such mobile content and to do so in a multiplatform way, while enabling the revolutionary mobile hardware features where available.

Thus our goal in this brief analysis is to provide a competent unified multiplatform design suggestion for delivering content on a wide range of devices (mobile, desktop or something else), and to discuss four major challenges (that such an integrated design poses) and their four major solutions: faceting, accessibility, synchronization, security.

5.1. Faceting

The first and most important concept is faceting. A SCORM compliant multiplatform enabled Learning Management System should provide means to register a course with multiple "facets" or modules, all packaged according to SCORM standard packaging and sharing the same imsmanifest.xml structural backbone. A content facet should be "tagged" according to the intended delivery platform. This means that for a standard course we have a default web facet, which is to be delivered on desktop browsers in a web-based provided SCORM player. Additionally the content developer may provide a facet to enable this course to run on Android hardware, another facet for running on iOS hardware or yet another one enabling it to run in WML on Linux embedded devices such as set-top boxes (provided this could be useful). These facets provide essential content specifically designed to run on particular platforms (such as using the gyro hardware on an iPhone or display resolution adapted to mobile devices or search integration within the mobile OS) [7].

By means of faceting the content can be delivered to multiple platforms and a centralized equivalence between each of these deliverables can be established. This way a course can be divided in multiple sessions, where not all sessions must be taken on the same device or platform, and eventually all sessions are synchronized in the LMS as if delivered to a single platform. Furthermore, faceted content can be developed to enable mixed multi-device sessions however the granularity of the session-to-device distribution must be quantified in SCOs and no lower.

The engaging of other than web-based player content should be done by platform specific SCORM player implementations, which perform the role of clients and also allow security measures integration (this way the player can gain access to the mobile device features, unlike a mobile browser even if using HTML5). Content can be developed using HTML5, Flash, video, audio and JavaScript while using 3rd Party APIs to provide native device functionality (more on this in the **'Player Flavours'** section). This solution opens the gates to native functionalities for mobile content (which is custom tailored to the platform and, in this way, takes full advantage of mobile device features such as GPS, camera, microphone, gestures, accelerometer), but also enables a much more secure and accessible experience, as we shall see next.

5.2. Accessibility

Unfortunately, the field of mobile applications and devices lacks sufficient effort towards their accessibility. Ignoring the accessibility of mobile applications and devices affects in the largest extent the slightly impaired users, who represent a relatively large segment of the market [8]. The mobile SCORM Player implementation provided by the LMS should provide it's own accessibility enhancements where this is possible, to ensure the delivery of content reaches a larger audience.

Given the nature of mobile devices, here are some of their highly expected characteristics: screen sizes are small, text is not always clearly visible, speakers are not always loud enough (or the student may have hearing impairments altogether), capacitive displays don't go well with shaking hands or certain UI elements could be too small for some fingers. That is why content developers must pay great consideration when developing mobile content. By using a native SCORM Player for a device it is easy to provide platform accessibility features integration for content or even specific accessibility options designed by the Player itself. These may include but not be limited to: font sizes, color themes, text-to-speech, voice recognition input, hand shaking and fat-fingers countermeasures.

5.3. Synchronisation

The desired scenario when working with a mobile enabled LMS, such as the one we're trying to explain here, is to deliver the same course to multiple devices (or more correctly to deploy a course's facets to multiple delivery clients) and then track the progress. The SCORM Player should be able to perform offline as well, thus "downloading" courses to engage offline when Internet connectivity is available. Now, if the user engages a SCORM unit (course, package, SCO or activity) on one platform and has no Internet connectivity, the native SCORM Player takes on the role to buffer all the outgoing communication in a secure encrypted way and also keeping an accurate datamodel [9] on the client. When a connection becomes available all such changes would ideally be uploaded to the LMS. This is called synchronization. But the problems appear when multiple devices perform different progress on the same unit and then upload conflicting progress.

The solution is to use a "version control" like system, in which we introduce the concept of check out and check in so that check out means downloading a content for offline execution and check in means uploading all the offline buffered data model. By using such a system, a lock can be established, so that whenever a course is downloaded for offline engaging on a certain device, no engaging on any other device or platform may occur until the lock is lifted.

Variations may be used, such as different types of locks (download for preview only), or even use a versioning system in which the LMS alerts the user about conflicting progress and asks which one to keep as relevant.

Keeping an offline data model for each course may be costly and there is a trade-off between offline playing capability and data plan bills, but wireless internet mobile devices are becoming more and more popular and also compression algorithms may be used to transfer the data model (encrypted compression for better security) to and from the device.

5.4. Security

The last topic leads us to a brief analysis of security issues concerning mobile devices. It is no news that SCORM is not a standard implementing any security related specifications. Using content to assess sensitive learning objectives (requiring results fraud prevention), without taking this issue into account, is a great risk [10]. A SCORM compliant LMS alone is not going to do much. Since it's a very well know issue, vendors such as Questionmark and many LMS vendors provide encryption mechanisms for sensitive content assessments.

If an institution is using iPads to deliver content to its employees and it delivers assessment tests to determine their level of qualification, a secure connection must be provided. This must be taken into account by any mLMS and it is suggested that encryption extensions be used.

SCORM does not provide a "proctoring" mechanism. However in this design that uses a dedicated SCORM Player, it is all too possible to implement timers, exposure verification mechanisms (to prevent or notify if the user is not viewing the content) and effectively preventing many possible cheating methods though not all. Camera screenshots taken periodically during a test (under the agreement of the test taker) could ensure a mechanism to prevent proxy test taker fraud [11].

Security is a major concern for all mLMS users (institutions, companies and their employees or students), but this section has highlighted the most important potential solutions to the common security concerns.

5.5. Player Flavours

As mentioned already, there is no other apparent way to enable content to take advantage of device specific hardware capabilities, unless through a native SCORM Player. The question is: How does a SCORM abiding content use such capabilities unless it's a native app itself?

And the all-pervading answer is: through the player. The implementation acts as a browser container (and to some large extent, it is), which provides not only the SCORM API object but also a proprietary (due to lack of standards) yet open API. Such an API can be invoked and used in an extremely easy fashion by any content (be it Flash or HTML with JavaScript) and can provide an interface to the device's microphone, camera, or any other specific hardware in an easy to use fashion. Also an LMS implementer can provide a similar API for each platform, including one for the old desktop browser-based experience.

This makes the SCORM Player not only just a browser but a complete container which takes it upon itself to synchronize content, communicate with the LMS (in a safe encrypted way), buffer offline datamodel operations, provide extended APIs for native hardware extra functionalities and last, but not least, it does all these and maybe even more in a full SCORM compatible way.

VI. CONCLUSIONS

It is quite obvious that mLearning is the future and there is a striking advantage in providing standard (SCORM) complying mobile content with the full-fledged benefits of innovative mobile hardware.

In conclusion, we propose a number of achievable directions and potential solutions to a better integration and a more streamlined mLearning. By using native SCORM Players and an LMS designed to support faceting we assert the following benefits and address the key areas of concern: accessibility, security, synchronization and functionality.

This mLMS design does not throw away any existing SCORM content. Instead, by adding mobile facets to existing courses, it enhances the effectiveness and widens the experience into the mobile world of smart phones and tablets, while keeping everything together nicely synchronized in one single place.

Developing a mobile LMS is not a solution, but enabling a SCORM compatible Learning Management System for mobile delivery is the best approach to prevent limiting the extension to additional platforms and to remain open to future unforeseeable requirements.

References

[1] eLearning Guild 360 Report, 2007, http://www.elearningguild.com

[2] Weissberger, A.J. Exponential Growth in M2M Market Dependent on Important Network Enhancements http://viodi.com/2010/10/07/exponential-growth-in-m2m-market-dependent-on-important-network-enhancements/

[3] iTunes U and iAuthor may Impact Sooner than you think http://www.enyotalearning.com/blog/2012/02/21/itunes-u-and-iauthor-may-impact-mlearning-sooner-than-you-think/

[4] iTunes U Site Guidelines, http://images.apple.com/support/itunes-u/docs/iTunesU_Site_Guidelines_011912.pdf

[5] Haag, J. – From eLearning to mLearning: The Effectiveness of Mobile Course Delivery, Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2011

[6] Adobe roadmap for Flash runtimes http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/flashplatform/whitepapers/flash-runtimes-roadmap.pdf

[7] 6 Problems with Mobiles in Education and 6 Solutions. A viewpoint as infographic. http://savingict.posterous.com/6-problems-with-mobiles-in-education-and-6-so

[8] Vystrcil J., Mikovec Z., Fronc M., Drbalek J. Accessibility Issues Simulation For Mobile Applications, AEGIS Workshop and International Conference - "Accessibility Reaching Everywhere", 29-30 November 2011, Brussels, Belgium

[9] SCORM Users Guide for Programmers, ADL SCORM Documents Repository, http://www.adlnet.gov/wp-content/uploads/2011/12/SCORM_Users_Guide_for_Programmers.pdf

[10] SCORM Security – Some Perspective [http://scorm.com/blog/2009/04/scorm-security-some-perspective]

[11] Frank A.J. - Can the Online Proctor be Reliably Computerized? - Proceedings of the 2010 International Conference on e-Business (ICE-B)