Developing a Mobile Learning Environment: An Axiomatic Approach

Walter Rodriguez
Ngan Nguyen-Huynh
Arnold Fernandez
James Royal
Victor Fernandez
Florida Gulf Coast University
USA

ABSTRACT

A new mobile environment for learning has been designed via an axiomatic approach. And by simultaneously designing both tools (software) and processes (pedagogy), the resulting environment matches the functional requirements of the instructional program. *This paper describes the axioms established for mobile learning as well as development of the mobile computing environment. The paper discusses the developmental evolution and system architecture as well as the requirements of the portable training programs being offered via this new system. Apps are designed to connect learners, instructors and practitioners as well as to facilitate collaborative learning from a variety of mobile devices, anywhere in the World. Future content, apps and systems development will connect the physical and virtual environments, in order to truly enhance the mobile learning experience for people on the move.*

INTRODUCTION

> *The main drivers of innovation in higher education are not simply a function of what is technologically possible; they are—or should be—a function of pedagogically sound and cost–effective strategies that advance our institutional missions in ways that best serve our students, are fair to our faculty, and advance the interests of our communities (CIC, 2013.)*

This research seeks to bridge the theoretical gap between "mobile computing" and "mobile learning," by simultaneously designing both tools (i.e., software apps and systems) and processes (i.e., instructional methods and pedagogy). By parallelizing functional requirements, the research shows how mobile software applications can be designed to meet the unique pedagogical requirements of adaptive, portable, individualized, certification training programs for ever moving personnel. This process consist of concurrently (1) analyzing the learning needs, learning styles, and cost constraints of the learners being served (i.e., learners on the move) and (2) developing pedagogy, methods, applications and systems in response to the factors or functional requirements.

For instance, a recent survey (ITI, 2013) concluded that international business professionals: (a) are more mobile than the rest of the population; (b) have very limited time to study; (c) have limited access to PCs at home; and (d) the cost and time constraints of the tuition assistance (TA) or Financial Assistance (FA) programs is a major factor in their decision to embark in a program of studies. This led the researchers to design short-duration (but high-demand) career certification programs. In addition, it was determined that a mobile learning environment had to be designed for accessibility on popular mobile devices. Further, understanding that people are often preoccupied with challenges should also be an important consideration when designing both tools and pedagogy. After extensive discussions, the team concluded that the apps and lessons had to be very engaging and designed around short-term study time and reflection (i.e., 10 to 15 minutes), rather than prolonged seat-ins (i.e., hours of continuous study).

As global families are forced to become more mobile to find jobs (and as cost of education rises), both ubiquitous learning and ubiquitous computing gains in relevance in this learning space. Mobile learning researchers and mobile computing developers must work more closely. By parallelizing pedagogy and technology requirements, this paper shows how mobile software applications can be designed to meet the unique instructional requirements of adaptive, portable, individualized, certification training programs for ever moving personnel.

The learning space (school, college, and university) and workplace (business, industry, and government) are rapidly overcoming the barriers of distance, time and cost (Rodriguez, W. et al., 2010). Mobile software apps are becoming a key enabler of learning from anywhere, since there are now over six billion mobile phone subscribers. To wit, for “every person who access the Internet from a computer, two do so from a mobile device” (UNESCO, 2013). Therefore, ubiquitous mobile learning offers the greatest potential for increasing “the boundaries of anytime and anywhere learning for students” (Yan et al, 2006). But, of course, mobile learning requires a pedagogically sound and cost-effective “enabling technology” designed for that purpose (CIS, 2013; Duderstant, 2011; McGreal, 2012; Wagner, 2005).

In a “Pedagogical Framework for Mobile Learning,” Park (2011), extensively discussed the evolution of mobile learning while “on the move” (Beckmann, 2010) using wireless devices—including smartphones, tablets (Kukulska-Hulme & Traxler, 2005) as well as technological attributes and pedagogical affordances.

AN AXIOMATIC APPROACH

The development of computer supported learning environments should follow some basic axioms (Suh, 2001; Rodriguez, 2011). Axioms are self-evident truths (*Encyclopedia Britannica Online, 2011*). Despite the initial technological orientation of the research, early on, the researchers settled on:

Axiom 1: Technology must follow pedagogy and not all the way around.

This axiom echoes the design principle “form follow function,” as attributed to Sullivan (1896). In this case, the functional requirements of portable training programs being offered need to be established before developing apps or the learning system to deliver those programs. Further, apps must be designed to: (1) connect learners, instructors and practitioners; and (2) facilitate collaborative learning from a variety of mobile devices, anywhere in the World, since global personnel and their families are very mobile. Finally, future learning apps must be developed to connect the physical and virtual environments, in order to truly enhance the ubiquitous learning experience for global personnel on the move.

Park (2011) was able to illustrate the conceptual shifts from e-learning to mobile learning where at some point the devices and systems may disappear. While we arrive at a future where learning may be embedded into everything, it makes sense to improve mobile learning by concurrently designing tools (mobile devices) and learning processes (pedagogy, strategies and methodologies) to meet the desired learning objectives. The next paragraphs present the evolution of one of such a system, collectively known as "Coursewell," as well as the enabling technologies used to connect students (knowledge seekers) directly to instructors (knowledge providers) and practitioners (business/industry) via natively-designed applications and mobile devices.

The first programmatic decisions for the certification training program was to determine the objectives or functional requirements of the program, since they will have a critical impact on the resulting instructional content and the corresponding systems and software applications. The influence of these early decisions over the entire process is profound. Further, the course designers must clearly define the project's functional requirements, or FRs, since they will greatly influence other design decisions that follow. This axiom is summarized as:

Axiom 2: Early Curricular Decisions Greatly Affect the Final Design Outcome

Additional axioms have been described by Suh (2001), namely: (1) the Independence Axiom and (2) the Information Axiom. The first axiom states that the systems' Functional Requirement (FRs) must be independently satisfied, while the second axiom states that designers should minimize the information content (Rodriguez, 2011). Although within the context of learning, it should be mentioned that the individual context should be integrated with the collective contextual experience. To wit, individually satisfying a requirement may be fine for software development, but this may not benefit the gestalt of the learning experience. Understanding this precept, the general axiom can still be captured as:

Axiom 3: Functional Requirements Must be Independently Satisfied

After determining the students' needs (i.e., access lessons on the move, adaptability, short attention spans), the research established a series of functional requirements (FRs), as follows.

a. Course content and offerings must adapt to the student learning style.
b. Provide various ways to learn the material (i.e., games, forums, etc.)
c. Facilitate access to collaborative discussion forums and project in order to increase student collaboration with each other.
d. Software apps must be able to assess each of the learning objectives, as soon as the students complete a lesson.
e. Content must be divided in to small manageable units for short attention spans, based on specific learning objectives (LOs)
f. System/apps must provide immediate feedback to the learner.

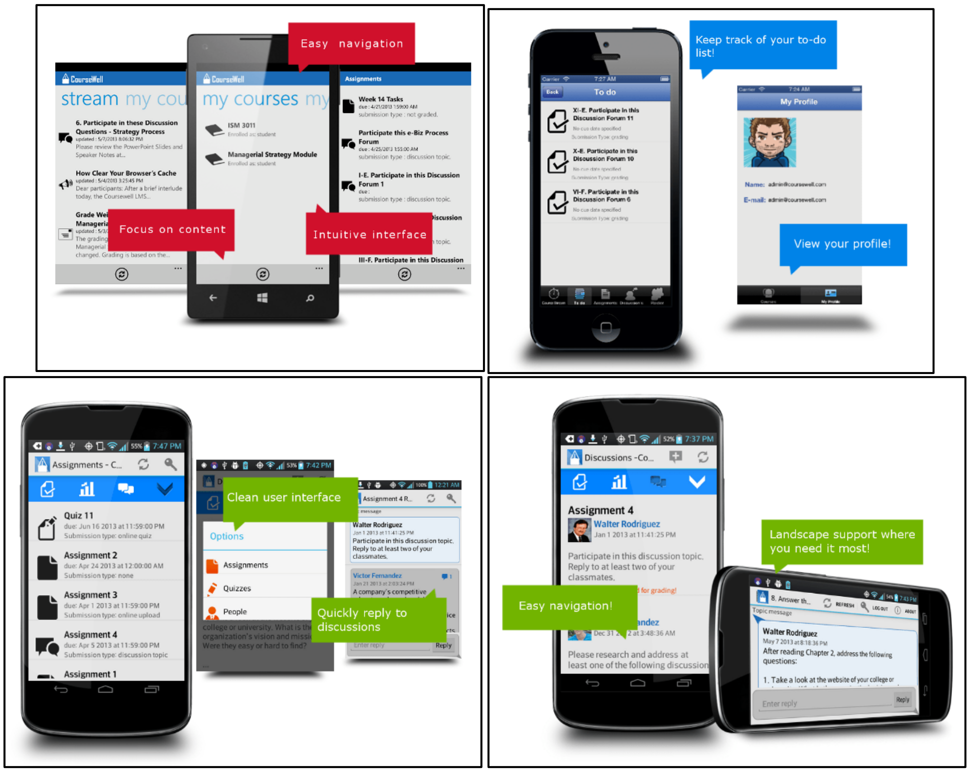
Before discussing the mobile computing development effort, let's briefly describe the adaptive learning pedagogy used in the certification training program. Adaptive learning is a computer-based instructional methodology in which users (students, faculty and practitioners) are able to learn and teach based on their learning abilities, needs and wants (Wikipedia, 2013). An adaptive system is capable to modify lessons depending on difficulty (or interest) and provide adequate time to complete. Generally, using cognitive scaffolding (Fernandez, 2003), users are presented short lessons or modules. The answers to the subject questions are evaluated by the system to provide formative and summative assessment. Quizzes and tests, among other formative learning activities, are pre-designed for specific learning objectives. The adaptive learning program is focused on designing and delivering fully-adaptable certification training courses and modules to respond to students' needs and demands.

FUNCTIONAL REQUIREMENTS

The enabling technology has been in development for about 18 months. And it builds upon robust Open Source Learning Management System (LMS) technology. The project was initiated for learning through mobile devices. And, early on, it was decided to have two teams of student researchers working concurrently. One team of management students researched the needs of the certification training participants and defined the functional requirements, while a group of computer science/software engineering students started working on the software development architecture with a clear understanding of the basic axioms and functional requirements discussed above.

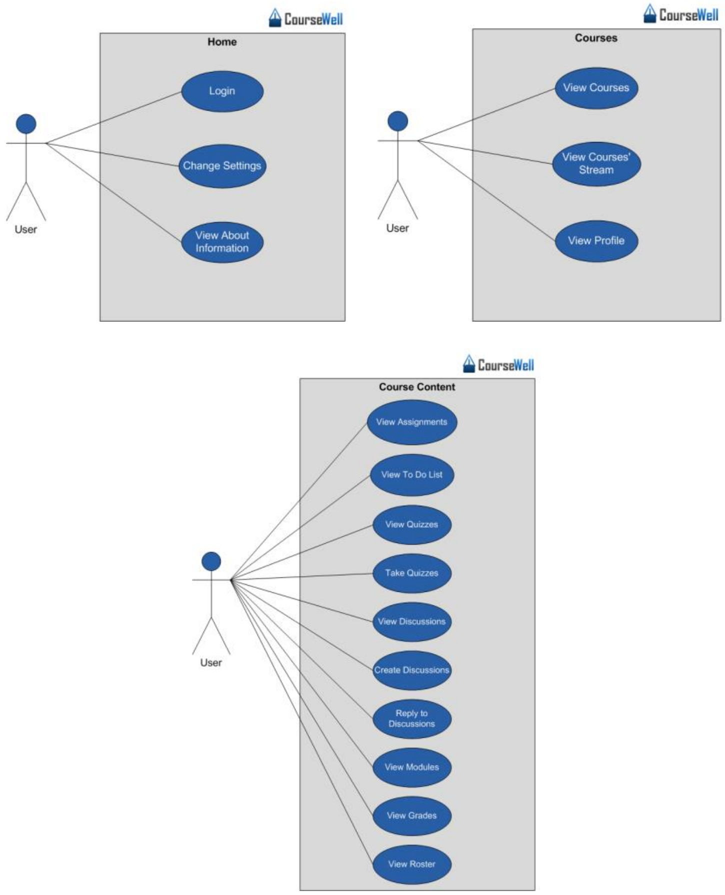
Figure 1 shows the features of the enabling technology, named "Coursewell." Essentially, it consists of natively-design apps and systems that have been developed specifically for enhancing the forum discussions and interaction among certification program participants.

Figure 1: User Interface and Features.



Both the business student and computing groups met with the principal investigator to review the objectives, functional requirements and program features on a weekly basis. The analysis started by drawing the use case diagrams illustrated in Figure 2. These diagrams are a model of the proposed user's interactions with the system. And they depict the specifications of a use case, i.e., list of steps, defining interactions between a role (actor) and a system (Jacobson et al, 2011). The top diagram on the left specifies that the user will be able to login and change settings, while the diagram on the right specifies that the user will be able to view courses, course stream and profile, after logging on. The bottom shows content and features.

Figure 2: Use-Case Diagrams.



As illustrated in Figure 2, the app and system users (learners on the move) are able to view the assignments posted by the instructors and practitioners as well as short "to-do" lists to keep the student attention and focus on the learning tasks at hand. In addition, students are able to view and take quizzes targeted to assess each of the short lessons at hand, after reading and engaging in the posted content (i.e., brief articles, short videos, simulations, situations, problems). More importantly, users are able to participate (reply or post) on discussion forums created by the instructors in collaboration with practitioners. Participants are also able to view and spend as much time as needed in the learning modules (i.e., articles, links, videos, annotated slides and so on). Students can view the class roster and directly contact the instructor, practitioners and their peers. Finally, they can view their grades, as soon as posted. Finally, within the Home-page's

“About” menu, users are able to provide feedback and suggestions to the software developers as well as rate the application.

SYSTEM ARCHITECTURE

Rather than the approaches followed by many Massively Open Online Course (MOOC) providers (Yuan & Powell, 2013,) Coursewell apps and system offer a personalized learning experience with an experienced instructor at the helm. In all courses, the Coursewell team involves students, faculty and industry practitioners.

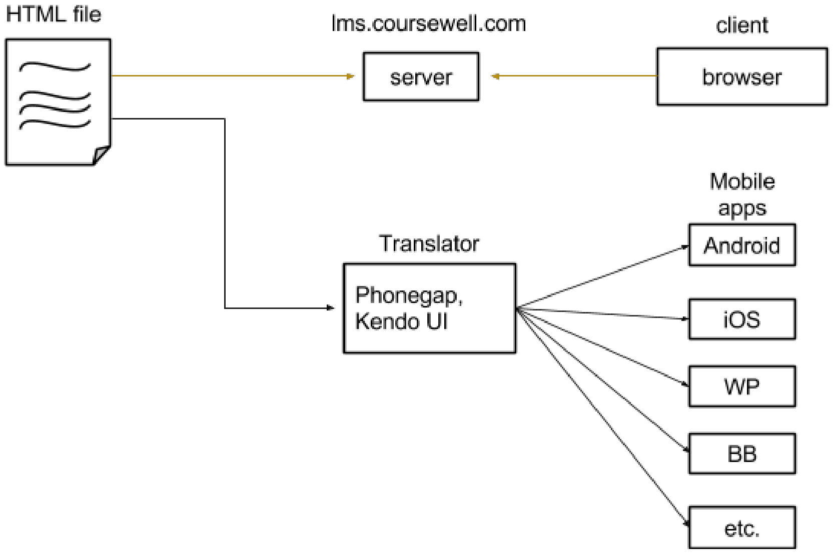
The following discussion is aimed at capturing the analysis and systems development of the Coursewell architecture. And it is organized to facilitate the project understanding for future developers. To increase the content relevance to ubiquitous learning, the authors discuss the values and the pros of the centralization of resources offered by the Open Systems LMS architectural paradigm.

The original Coursewell team of students finished the essential development tasks and the project took on a new goal of creating a Web App. So far, the project presence on the Internet has been formed through the website at www.coursewell.com using open source web development software (WordPress, 2013). Most of the Learning Management System (LMS) server development was performed by computer science research assistants. And it’s accessible to global learners via lms.coursewell.com. Readers and course developers (instructors) may request free access by writing to admin@coursewell.com. The presence among mobile device users has been advanced through three separate developments: (1) a naively-designed mobile application for Windows Phone (WP) 8 operating system; (2) a naively-designed mobile application for Android platform; and (3) a naively-designed developed a mobile application for iOS platform.

After the initial research team completed their efforts, the idea of having a centralized mobile app was proposed. Thus now, the goal is to centralize Coursewell development into a Web App, that is, an app that is run on the server and accessed through a web browser. This way, development cost can be minimized and Coursewell accessibility can be maximized. The initial supposition entailed working with HTML5 technology to make www.coursewell.com more mobile friendly. The initial Coursewell team’s goal was to use HTML5 technology and built mobile applications for all mobile platforms. Figure 3 shows the bifurcation of these implementations.

Both of these implementations have advantages and disadvantages. When the developed code is hosted on the server, the performance of the code may not be as good as code running natively on a device. However, this is subject to: (1) the optimization of the translator and (2) how well is the code written (with HTML) and translated into native code. Researchers discussed the differences in implementation in order to meet the concurrent design efforts (pedagogy and technology). And it was decided to develop a mobile interface for mobile device (i.e. smartphone, tablets). And work on a mobile User Interface (UI) through web development to match the FRs. That is, write code, host it on the server, and display the UI through HTTP request (from the browser) to facilitate the desired pedagogy (i.e., short attention spans; etc.)

Figure 3: Coursewell Mobile Access Implementation.



One advantage of developing the mobile UI, as proposed, is that it fills a niche. The Open Source LMS (or Canvas) was originally designed by Instructure™. And, at the moment of this writing, Instructure Inc. has released their mobile application for Android and iOS platform. However, they have not optimized their web version when browsed with a mobile device. Originally, the time allocated to create a prototype for the Web App was three months. As of this date, the Web App development has passed half of that time frame. Due to the explorative nature of this project, the time required for a prototype was extended another three months.

Currently, Coursewell website implementation composed of few separate components. Two components will be discussed here: the general website www.coursewell.com and the Open Source LMS. The current setup and integration of www.coursewell.com is shown in Figure 4. There is a difference between the implementation of www.coursewell.com and lms.coursewell.com even though they may seem to be one entity. Understanding the different implementations can clear ambiguity.

Beside the main LMS hosted with Amazon Web Service (AWS), there is also a development and testing server hosted within the hosting university domain. The research team configured the development server to be accessed through Secure Shell (SSH). And development of new Coursewell Mobile Website is currently taking place on that shell.

Figure 4: CourseWell Server and Components Interaction.

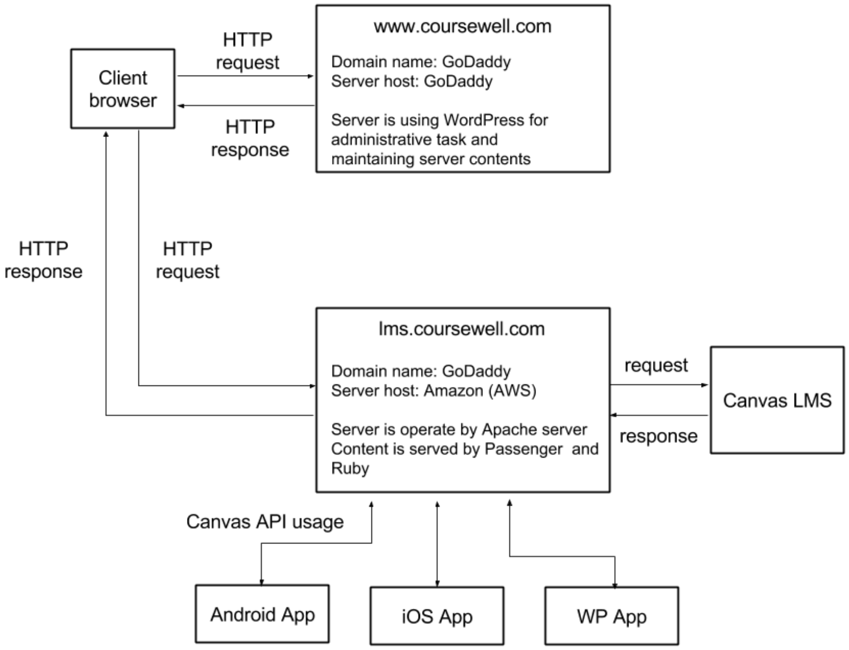
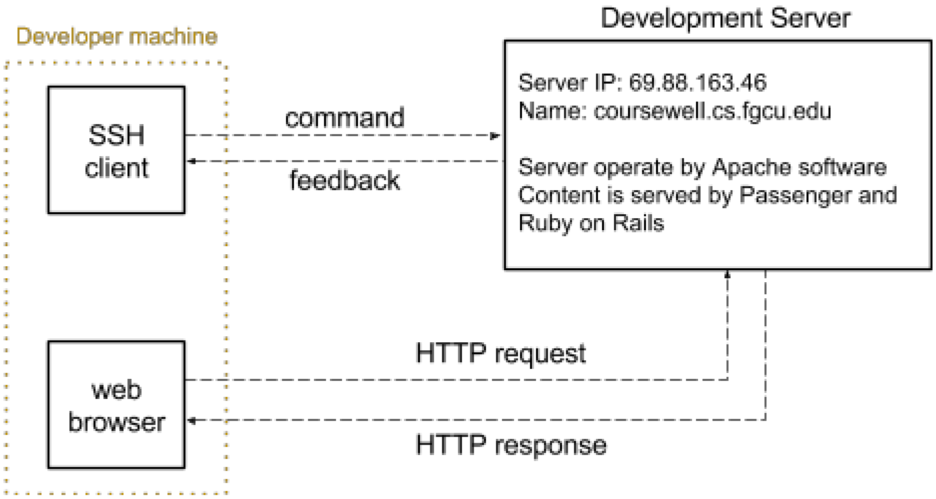


Figure 5 provides a visualization of the development process. Since the previous LMS version, the development server at coursewell.cs.fgcu.edu has been updated to match the setup at lms.coursewell.com. And the current state of the development server is non-operational. Due to new updates, the different components needed to run the LMS are not working together. Coursewell contains setup information for both of the following servers: coursewell.cs.fgcu.edu and lms.coursewell.com. The Canvas LMS is a massive collection of code. There are, roughly estimated, about 1 million lines of code. To understand and effectively developed features for Canvas LMS, an IDE should be used. Currently the development setup use Eclipse IDE and plugins (Aptana, 2013).

Figure 5: Development Setup.



It should be mentioned that in addition to the main Open Source LMS hosted with Amazon Web Service (AWS), there the developers implemented development and testing server hosted within the university campus. This test server was configured as the development server to be accessed through Secure Shell (SSH). Development of Coursewell Mobile Website is currently taking place on that server. Figure 5 provides a visualization of the development setup.

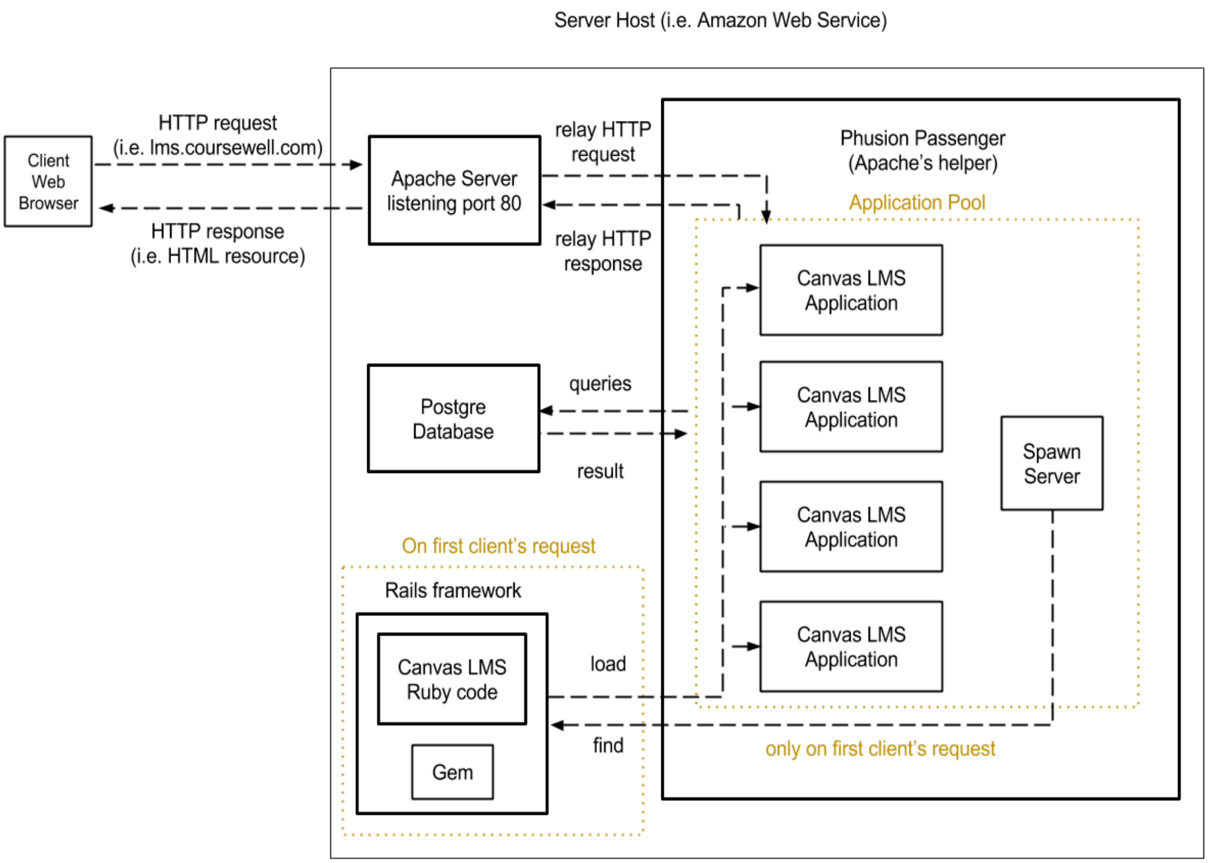
Since the previous version, the development server at coursewell.cs.fgcu.edu has been updated to match the setup at lms.coursewell.com. The current state of the development server is non-operational. Due to new Open LMS updates, the different components needed to run the LMS are not working together. Coursewell documentation (2013) link in the References contains setup information for both of these servers, namely, coursewell.cs.fgcu.edu and lms.coursewell.com.

It should be noted that the Open LMS is a massive collection of code and there are roughly an estimated about 1 million lines of code. To understand and effectively developed features for this Open LMS, an Integrated Development Environment or IDE should be used. Currently the development setup uses Eclipse IDE and plugins (Aptana, 2013). For instance, the development setup for a developer with Windows Operating Systems is as follows:

- Install Eclipse 3.6 or above,
- Install Aptana Plugin
- ○ Help > Install new software : http://download.aptana.com/studio3/plugin/install
- Download Canvas LMS source code (Git Website) - alternatively, use Git to fork.
- Create new project
- Import Open Source LMS source code (as File System)

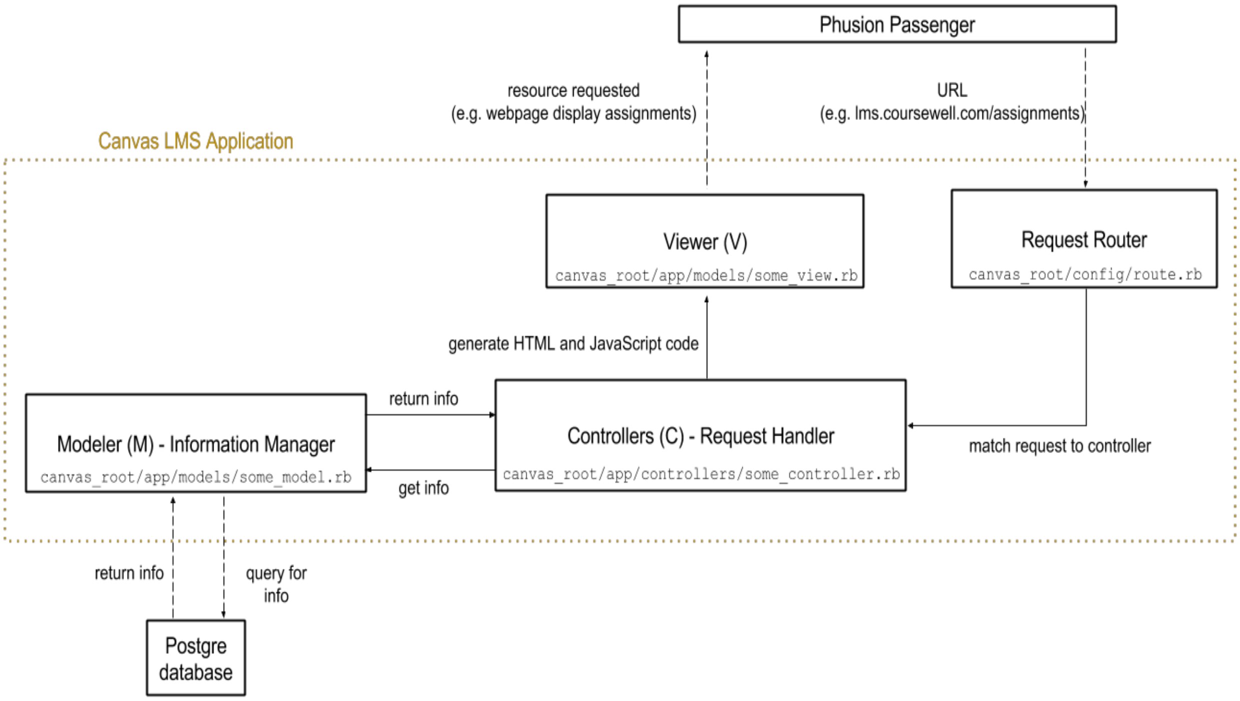
There are a few software components within Coursewell development project. And let’s now focus on the components located on the server that make the Open LMS operational. Figure 6 provides an overview of how these components operate together to serve content upon client’s request.

Figure 6: Coursewell Components Overview.



The Open Source LMS (or CANVAS) is a simplified entity. When analyzed, its operation required the integration of multiple advanced web technologies. The website lms.coursewell.com, or any website produced by a server hosting this LMS (e.g., coursewell.cs.fgcu.edu), is dynamically generated by code written in Ruby. All functionality of this LMS is written in Ruby language and operates through Rails framework. Each webpage (e.g., Course, Assignments, Grades, Calendar) within the LMS is generated dynamically by the Canvas Application (CA). In Figure 6, after a user's HTTP request is relayed from Apache2 server to the CA, the request will be parsed by CA, then routed to a corresponding controller and eventually generate a display. This is how the LMS is able to perform many "functionalities" through the different routing options and corresponding controllers. Figure 7 visualizes this operation in more detail.

Figure 7: Rails MVC Perspective of Canvas Open Source LMS.


Phusion Passenger
resource requested
(e.g. webpage display assignments)
URL
(e.g. lms.coursewell.com/assignments)
Canvas LMS Application
Viewer (V)
canvas_root/app/models/some_view.rb
Request Router
canvas_root/config/route.rb
generate HTML and JavaScript code
return info
Modeler (M) - Information Manager
canvas_root/app/models/some_model.rb
Controllers (C) - Request Handler
canvas_root/app/controllers/some_controller.rb
get info
match request to controller
return info
query for
info
Postgre
database

The role of the Apache Server within the LMS is to listen to and validate HTTP requests. Apache also maintains the domain's name of the current host (the computer) when it is accessed from the Internet. This component's functionality can be abstracted from this research project and a Mobile Web App developer would only need to understand HTTP requests from the Internet are received by Apache server, and then relayed to Phusion Passenger (2013) software.

The role of Phusion Passenger (2013) is to aid Apache server (software), since the Apache server was not designed to execute web app. A few factors are involved with the integration between Apache server and Apache. Detailed documentation can be found in *Phusion Passenger relationship with Apache (*Phusion Passanger, 2013). In order for Phusion Passenger to run Ruby on Rails application (i.e. Canvas Open Source LMS), the two entities must be integrated. Detailed documentation of this integration can be found in *Phusion Passenger relation with with Ruby and Gems*. During the initialization of a Ruby on Rails application, Passenger must load this application onto its application pool (a collection of redundant copies for performance purposes). For further discussion, the reader may seek the *Phusion Passenger (2013) architecture* which provides an overview how Phusion Passenger was designed. Rails framework is one way of organizing Ruby code. This framework could also be understood as a standard of writing application. Through this standard, functionalities and integration can be maximized through reused of the codes (the codes that define ways certain functionalities must be implemented).

It should be mentioned that Rails (2013) follows the Model-View-Control (MVC) architecture. When source code is designed within this architecture, it can take advantage of Rails function that route URL to certain Controller without dealing with the server that received the HTTP request. And it can take advantage of Rails function that build database relationship through Model without dealing with database-specific commands. *Rails for Zombie* can provide a tutorial for hands-on experience with Rails' MVC architecture. As seen above, the language within Canvas Open Source LMS is mainly Ruby. Although there are a few HTML and CSS source code lines, most of the software is written in Ruby. In order to extend functionality of the Open Source Canvas LMS, the developer must know Ruby. Also, in order for Canvas LMS to function properly, it must have all the dependent Gems installed. This can be very challenging when Canvas LMS has many released updates, and Gems' untamed nature. Gems are like a library in languages such as C++ or packages in Java, except that it's not as stable because of the fast development nature of web technology and constant update to codes.

VALIDATION OF LEARNING ENVIRONMENT ON TARGET AUDIENCE

During fall 2014 semester 14 students beta-tested the LMS and apps while taking the online ISM 3011 Information Systems course taught by the first author and his assistants at Florida Gulf Coast University's Lutgert College of Business. This hybrid (online/on-campus) course consisted of 16 online modules and it culminated with a live presentation the last day of classes. Upon completion of all online modules and presentation, students were asked to complete an electronic survey measuring their level of satisfaction with the system in seven factors below. Of the 14 students 10 responded to the survey, the results of which are listed in Table 1. It was concluded that significant improvement is needed on the apps' discussion-forums (i.e. friendlier GUI), while the LMS performed well.

Table 1: Students Reported Satisfaction Using the LMS and Apps.

	Extremely Satisfied	Very Satisfied	Satisfied	Dissatisfied	Very Dissatisfied
Overall LMS Performance	40%	40%	10%	10%	0%
Apps Performance	10%	40%	20%	30%	0%
Graphical User Interface	20%	10%	40%	30%	0%
Uploading Files on LMS	30%	30%	40%	0%	0%
Downloading Files on LMS	20%	20%	40%	20%	0%
Discussion Forums on App	20%	20%	50%	10%	0%
Quizzes on LMS	10%	30%	50%	10%	0%

CONCLUSION

Concurrently designing the mobile learning pedagogy and the mobile learning computing environment can be very powerful for those willing to try this approach. Further, by having clear objectives and functional requirements, the required system integration allow the developers to simultaneously be developing the instructional program while writing source code more effectively within this project environment.

To wit, Coursewell adaptive mobile learning system offers a potential for truly mobile learning and has tremendous potential to avail learning to anyone with a smart phone or other mobile devices. As discussed, it overcomes the constraints of distance, time and cost experienced in traditional campus-based learning by providing the advantages of a mobile education system and apps that connects students, faculty and practitioners everywhere.

Coursewell adaptive, mobile collaboration learning system and applications are delivering valuable, individualized, quality education to business, industry and government by including teams of faculty and practitioners in the instruction and development of courses. In addition, Coursewell has been able to provide a way to support faculty/student research and scholarship endeavors.

Now that the main mobile-education software development efforts are completed, Coursewell enters the next developmental stage. For instance, Coursewell envisions evolving its technology into a more sophisticated multi-modal (Sankey et al., 2010) learning system that fully integrates the learning space with the workplace as well as the physical and virtual space. This notion is directly drawn from a concurrent research project titled uCollaborator---which involved several universities and consisted in developing a human-centered computing framework for improving teamwork and transforming the human-computer interaction experience for distributed teams (Rodriguez et al., 2010.)

As Coursewell evolves, it would be possible to connect teams of workers and learners that are geographically dispersed and interact at different times. Pursuant to the uCollaborator framework, it would achieve this goal through a multimodal team interaction interface realized through a reconfigurable open architecture. "Researchers at Florida Gulf Coast University, University of Central Florida, Florida Institute of Technology and Cognitive Information Science Laboratories have conceived this framework to integrate: (1) an intuitive speech- and video-centric multi-modal interface to augment more conventional methods (e.g., mouse, stylus and touch), (2) an open and reconfigurable architecture supporting information gathering, and (3) a machine intelligent approach to analysis and management of heterogeneous live and stored sensor data to support collaboration. The open architecture has five reconfigurable layers: input layer, process and storage layer, visualization layer, analysis layer, and decision layer." (Rodriguez et al., 2010).

When the above system is fully implemented, future learners would be able to draw data from sensors in the physical world (i.e., say vibration in bridges; or monitoring temperature or stress on objects or persons). Faculty and practitioners would be able to share data (numbers, images, videos, simulations) in real-time with students participating in a class. This data could be used to develop virtual simulations that are displayed on the Coursewell mobile learning environment. So, the class discussions could be based on both theory and practice. Finally, this research allowed a group of students in business and computer science/software engineering to work collaboratively. And the results were outstanding. Next step is to continuous to test and improve both content delivered and the associated technology, based on students' feedback.

REFERENCES

Aptana. (2013). Getting Started with Aptana. Accessed June 12, 2012 via URL at: http://www.aptana.com/products/studio3/getting_started

Beckmann, E. A. (2010). Learners on the move: Mobile modalities in development studies. *Distance Education*, 31(2), 159-173.

Committee on Institutional Cooperation (CIC). (2013). CIC Online Learning Collaboration: A Vision and Framework. Champaign, Illinois 61820. Retrieved from http://www.cic.net

Coursewell LMS documentation. (2013). Retrieved from: https://lms.coursewell.com/doc/api/

Duderstadt, J. J. (2011). A Master Plan for Higher Education in the Midwest: A Roadmapto the Future of the Nation's Heartland. Chicago Council on Global Affairs, Chicago, IL.

Encyclopedia Britannica Online (2013). Accessed November 10, 2013. Encyclopedia Britannica, Inc.

Fernandez, G. (2003). Cognitive scaffolding for a web-based adaptive learning environment, Advances in web-based learning. *ICWl 2003: International Conference on web-based learning*, Melbourne, 18-20 August 2003, Vol. 2783, pp. 12-20. Accessed October 2, 2013 from http://cat.inist.fr/?aModele=afficheN&cpsidt=15568725 .

FERPA (2012), Family Education Rights & Privacy Act. Retrieved from *www.ed.gov/policy/gen/reg/ferpa*

ITI. (2013). Survey of Military Spouses (Personal Communication with MedCerts LLC staff).

Jacobson I., Spence I., & Bittner K. (2011). Use Case 2.0: The Guide to Succeeding with Use Cases. IJI SA. Accessed at http://en.wikipedia.org/wiki/Use_case.

Kukulska-Hulme, A. (2007). Mobile usability in educational context: What have we learnt? *International Review of Research in Open and Distance Learning*, 8(2), 1-16.

McGreal, R. (2012). The need for open educational resources for ubiquitous learning. *Pervasive Computing and Communications Workshops* (PERCOM Workshops), 2012 IEEE International Conference, 679-684.

Park, Y. (2011). A Pedagogical Framework for Mobile Learning: Categorizing Educational Applications of Mobile Technologies into Four Types. *The International Review of Research in Open and Distance Learning*, 12(2), Retrieved from http://www.irrodl.org/index.php/irrodl/rt/printerFriendly/791/1699

Phusion. (2013). Phusion passenger relationship with Apache documentation. Retrieved from: http://www.modrails.com/documentation/Users%20guide%20Apache.html

Phusion. (2013). Phusion passenger relationship Ruby and dependent Gems documentation. Retrieved from: http://www.modrails.com/documentation/Users%20guide%20Apache.html#relationship_with_ruby

Phusion. (2013). Phusion Passenger Architecture documentation. Retrieved from: http://www.modrails.com/documentation/Architectural%20overview.html

Rails (2013). Rails for Zombie website, Retrieved from: http://railsforzombies.org

Rodriguez, W. (2000). Web Classroom of the Future: Integrating Course Management Software in a Java-based Environment, *Interactive Multimedia Electronic Journal of Computer Enhanced Learning*, 2(1), Retrieved from: http://www.imej.wfu.edu/articles/2000/1/07/index.asp

Rodriguez, W. (2011). An Axiomatic Approach to Digital Enterprise Design: From Conception to Realization. *Design Principles & Practices: An International Journal*, 5(5), accessed at http://www.Design-Journal.com.

Rodriguez, W., Fiore, S. M., De Welde, K., Carstens, D. S., & Kepuska, V. Z. (2010). Ubiquitous Collaboration (uC) Learning: Connecting the Physical and Virtual Worlds. *Ubiquitous Learning: An International Journal*, 2(2), Accessed at http://ijq.cgpublisher.com/ , ISSN 1835-9795

Sankey, M., Birch, D. & Gardiner, M. (2010). Engaging students through multimodal learning environments: The journey continues. ASCILITE, Retrieved from: http://www.ascilite.org.au/conferences/sydney10/procs/Sankey-full.pdf

Sullivan, L. H. (1896, March). The Tall Office Building Artistically Considered . *Lippincott's Magazine*, 403–409.

TheWhiteHouse.gov, (2013). Education: Knowledge and Skills for the Jobs of the Future. Retrieved from: http://www.whitehouse.gov/issues/education/higher-education.

UNESCO (2013). Working Paper Series on Mobile Learning. Retrieved from: http://www.unesco.org

Wagner, E. D. (2005). Enabling mobile learning. *EDUCAUSE Review*. 40, 3 (May/June 2005), 40-53. Retrieved from: http://www.educause.edu/er/erm05/erm0532.asp?bhcp=1

Wikipedia (2013). Adaptive Learning. Retrieved from: http://en.wikipedia.org/wiki/Adaptive_learning

WordPress (2013). Developer Documentation. Retrieved from http://codex.wordpress.org/Developer_Documentation

Yang, C., Tin, T., McGreal, R., Ally, M., & Coffey, S. (2006). The Athabasca University Mobile Library Project: Increasing the Boundaries of Anytime and Anywhere Learning for students. *IWCMC'06*, July 3–6, 2006, Vancouver, British Columbia, Canada.

Yuan, L., & Powell, S. (2013). MOOCs and Open Education: Implications in Higher Education. *JISC Centre for Educational Technology & Interoperability Standards (CETIS)*, Retrieved from: http://publications.cetis.ac.uk/wp-content/uploads/2013/03/MOOCs-and-Open-Education.pdf

APPENDIX

Coursewell LMS documentation, Retrieved from: https://lms.coursewell.com/doc/api/

Phusion Passenger relationship with Apache documentation webpage, Retrieved from: http://www.modrails.com/documentation/Users%20guide%20Apache.html

Phusion Passenger relationship Ruby and dependent Gems documentation, Retrieved from: http://www.modrails.com/documentation/Users%20guide%20Apache.html#relationship_with_ruby

Phusion Passenger Architecture documentation, Retrieved from: http://www.modrails.com/documentation/Architectural%20overview.html

Rails for Zombie website, Retrieved from: http://railsforzombies.org