A multi-disciplinary approach developing a mobile Curriculum Mapping App called iFUN

Michael Vallance, Hiroaki Murayama, David Sperling

Abstract — **Curriculum content and delivery during rapid technological advancements and socio-economic changes present complex challenges for universities worldwide. The paper summarizes a multi-disciplinary process developing an iOS App called iFUN which maps a university's curricula, thereby allowing course designers to determine curricula strengths, gaps, misalignment, and repetition. iFUN is being adopted to envision current and future curricula, and contributes to the pursuit of a more multi-disciplinary student education.**

Index Terms — **curriculum, design, programming, education**

I. INTRODUCTION

The impact of socio-economic changes and technological advancements is prompting many educational institutions to re-think the processes of planning, designing and delivering courses in Higher Education. The effect is a re-emergence of the importance of curriculum design [1].

Curriculum delivery presents many complex challenges: e.g. responding to learner needs; delivery of engaging and flexible learning experiences, and varying assessments. Like any act of design, curriculum design is usually based on some initial ideas. Yet because ideas about education are important politically and socially, the ideas that go into curriculum design are also usually politically and socially significant, or ideological. Policymakers, educators, businesses, researchers, parents, pressure groups and other stakeholders interested in a curriculum all attempt to generate support for their own particular ideas. For example, some academics do not share the enthusiasm for curricula designed to promote post-industrial work; some parents don't possess the same passion for children's participatory cultures; some policymakers disregard educational research that cite benefits of experiential learning over rote-learning and standardized testing. In other words, developing an education curriculum is a complex act of creative design.

To exacerbate the dilemma, the implementation of 1:1 technologies such as the iPad, iPhone, Android devices and laptops impacts upon academics, students and education managers. Despite resistance from some academic staff, it is imperative that a 21st century curriculum represents the development skills necessary for knowledge and information –based economies. This can only be achieved by adapting the way of teaching with the linking of syllabus content and new

forms of assessment: i.e. a multi-disciplinary curricula. What kind of approaches to curriculum design should we then be developing?

One approach is known as 'curriculum mapping' where stakeholders such as academic staff, students and community partners are engaged in a participatory development of a university's curricula. To facilitate access to a curricula and subsequently display links across courses in various disciplines, a mobile App was developed providing university course information and dynamically mapping curricula data based on end-users' search terms. This paper summarizes the development process.

II. CURRICULUM MAPPING

Curriculum Mapping identifies core skills, content and assessment in all taught subjects and highlights strengths, gaps, misalignment, and repetition [2]. It is recognized as a continuing process for orchestrating the scope and sequence of a curriculum to inspire coherence across grade levels, avoid unnecessary redundancies and provide teachers with timely feedback on curriculum implementation so that positive modifications can be made. The aim of this research project was therefore to design and develop a Curriculum Map mobile application which can digitally inform staff and students of course integration to provide a 'more informed' curricula.

TABLE 1. PLAN – DO – REFLECT

Dates	Technology goals	Education goals
PLAN April – June 2013	Input an updated syllabus to secure server using Java Eclipse.	
DO July – Dec 2013	Design a curriculum map of the syllabus for use in an iOS App. Use Node.js.	Keyword search for gaps and duplication in syllabus.
REFLECT Jan – March 2014		Trial some suggestions for modification to university syllabus, and see how these impact upon curricula. Obtain feedback from academic colleagues.

The research project comprised of 3 stages: PLAN – DO – REFLECT (see Table 1). The project has a technology focus and an education focus. An iOS App was subsequently

programmed so that data from a secure server can be illustrated as a curriculum map. The curriculum map can be updated in real-time by Administrator access to a secure server. Education researchers in the university's Center for Meta-learning (CML) can then analyze the resulting map for syllabus strengths, gaps, misalignment, and repetition.

III.   APPROACH

A.   Technology focused

Programming and development of an iOS application (using javascript) which allow users (after secure login) to amend courses on a dedicated secure server (using Java Eclipse) was implemented. Server-side programming and hosting support was provided by Solana Systems Inc., and iOS programming was undertaken by a Future University graduate student conversant with javascript and Node.js for App development. The usage procedure is as follows:

- User applies for a server-side account.
- User can update course theme, keywords, assessment, rate course relevance, and indicate other, essential linked courses.
- When completed, the curriculum map will refresh in the iOS App.
- The determining variables of the digital map will be primary (immediate course) keywords and secondary (associated courses) keywords, course relevance rating, and essential linked courses. This is programmed using Node.js: Node.js uses an event-driven I/O model for data-intensive real-time applications that run on distributed devices.
- The digital curriculum map is simultaneously updated in real-time taking into account the user's input.

B.   Education focused

We have designed, programmed and developed an iOS App for Future University's syllabus. The syllabus data initially had to be inputted manually by one person. However, this data is considered to be 'retrospective' data; in other words, it is the current syllabus (i.e. 2012-13) that was designed and confirmed in the previous year (i.e. 2012). Analysis of amended curricula was then undertaken by looking at repetition of course content, primary links between courses, gaps in content delivery, appropriateness of assessment type, opportunities for flexible pedagogy (e.g. the flipped classroom), and opportunities for new courses. This will be continually conducted by researchers and educators in Center for Meta-Learning (CML) at Future University Hakodate.

IV.   IMPLEMENTATION

The university's syllabus data was extracted from Excel and Word documents and added to an online Google Document. The data was arranged so that each column in the Google Document table represented possible variables that the App may retrieve. The document was shared with a translator who translated subject names and summaries from Japanese to English. The resulting document contained 184 subjects (the Rows) and 32 variables (the Columns).

The syllabus data was then imported to the Java Eclipse application so that the variables could be called upon when programming the iOS App on a dedicated laptop.

In order to design and develop the App, a program with objects and associated attributes needed to be prepared. The required information had to be specifically determined, and then represented diagrammatically so that the project leader, the programmer and the consultant were very clear about the project's outcomes and iOS App's focus. Figure 1 illustrates 3 of the 9 desired outcomes by the student user.
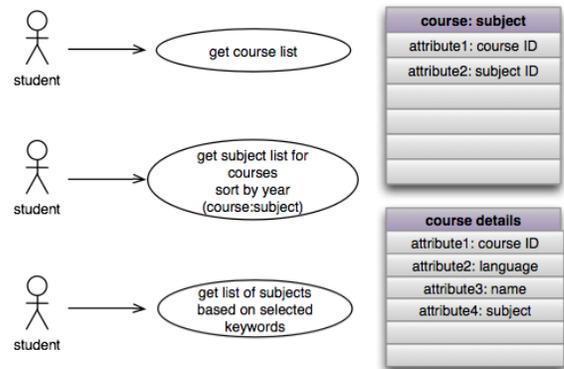

Fig. 1. App usage by end-user

Then the App's objects and variables needed to be determined, and the attributes needed to be specified. Once again, after much discussion, the agreed information was represented diagrammatically for clarity. The objects, variables and attributes were then represented in a model – controller - view (MCV) pattern (see Fig. 2). These three interconnected parts separates data within an application so that the data can be presented in different ways to the end-user. The model is the application data, the view is the output representation in multiple formats, and the controller accepts input and converts it to commands for the model or view.
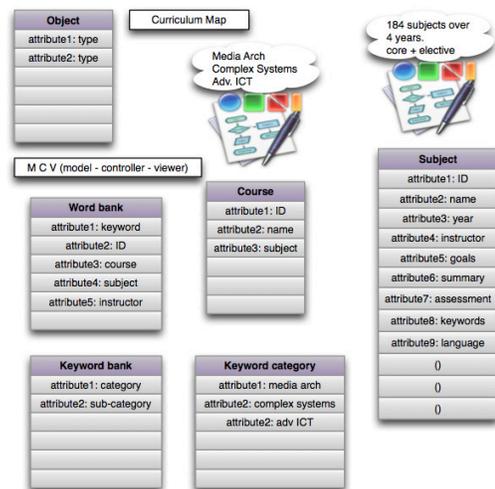

Fig. 2. Objects and Attributes

The syllabus data was then transferred to a dedicated server using the Java Eclipse program software. Eclipse retrieves specific data called a DAO (Data Access Object model). DAO's provide some specific data operations without exposing details of the database.

The App programming was undertaken using Appcelerator's Titanium Studio for Mac OSX software [3]. This has the advantage of developing one-time but being able to export to multiple platforms such as iOS and Android. The panes of the App were agreed to be Twitter, Links, Syllabus, Map and Settings. With the syllabus data now accessible from the server, the programmer could start to develop the App while other team members regularly updated the syllabus content. At the same time any updates to the syllabus could be undertaken by inputting data directly to the server thereby not interfering with the development of the programming of the App.

In time the App began to take shape with course lists per year, course information in English and course details in Japanese. It was decided to summarize the course details to the amount of text viewable in one iPhone screen size (see Fig. 3).
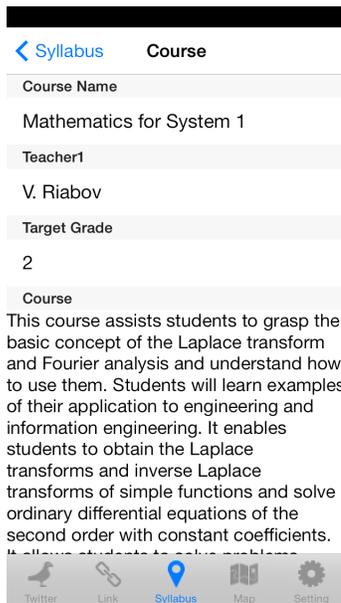

Fig. 3. Course details in English

The most challenging part though was programming the dynamic map. A similar concept was developed for an App called Wikilinks [4]. The project leader contacted the developers and found they were a team of programmers based in Switzerland who programmed their dynamic map using Python. Unfortunately, they were not available for contracting their skills, and we could not find a skilled Python programmer within our university. So we looked at Nodejs [5]. Nodejs is software that primarily builds network applications. It has been developed for data-intensive real-time applications that run across distributed devices. Similar to Titanium Studio, Nodejs ustilzes javascript as its scripting language.

After many hours over a 3-month period the first version was 'live' and could be tested on the iPhone Simulator on the programmer's MacBook. At this stage the Curriculum Map

became known as the Syllabus Map (see Fig. 4) due to a translation from English to Japanese where 'curriculum' and 'syllabus' are deemed synonymous. In English, 'curriculum' and 'syllabus' are distinctly different: 'curriculum' refers to course content, the teaching pedagogy and the form of assessment; 'syllabus' refers to the course and subjects' content only (i.e. what is being taught). However, given the relief at the result of a dynamic map we decided to dispense with semantics for the moment.
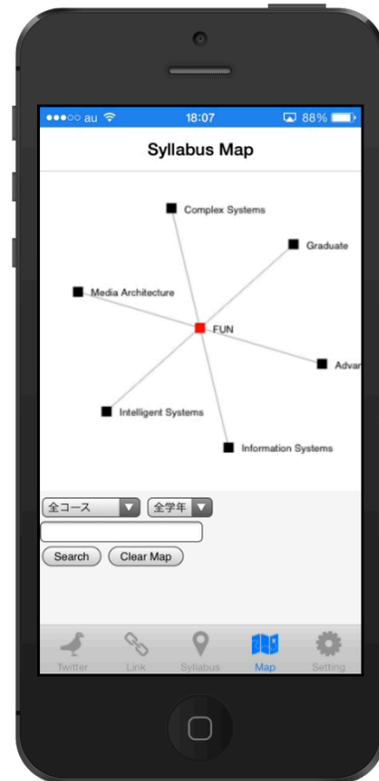
Fig. 4. Syllabus map

The syllabus data could then be searched by typing keywords in English (see Fig. 5) or Japanese. See Table 2 for an explanation.

TABLE 2. HOW TO SEARCH THE SYLLABUS

step 1: type a search term; e.g. java
step 2: decide course and year
step 3: click Search
step 4: let's look at the resulting map.
red node = keyword
orange node = primary course
black node = secondary course

At this stage the App was ready for beta testing. To distribute to staff and student volunteer beta-testers, the TestFlight [6] service was utilized for AdHoc distribution. In addition, the Titanium project files were uploaded to GitHub [7] for sharing between developers.

The App also needed a logo due to the interface design changes in Apple's iOS 7. A Future University design student

was commissioned and she created a logo consisting of the 3 letters of FUN plus 4 colors representing the 4 disciplines taught at Future University.

Fig. 5. Searching the syllabus

After feedback from beta-testers, a number of additional Twitter feeds and website links were added. It was also found that some keywords such as 'design' and 'create' occurred in so many subjects that the map became overwhelmed and there was not enough window space to show all the links. The subject keywords had to be curated to be more specific to each particular subject. In addition, the Graduate school courses were added to the server data and iFUN App program. As the initial stages in the determination of the variables were absolutely clear, adding the Graduate school information was seamless and quick.

Once the team had checked that all the variables were in place and accessible, the App was ready for distribution. An iTunes Connect account was created and the App uploaded for approval by Apple. Within less than a week, on Christmas Day 2013, the iFUN App was available for free on iTunes App Stores worldwide. The iFUN App can now be downloaded by typing the URL http://tinyurl.com/n3qa5pg

in a web browser, which will then re-direct to iFUN in the iTunes App Store.

V. CONCLUSION

The multi-disciplinary approach has led to a successful outcome which will prove beneficial to all current and future students and staff of the university. By tapping into the skills of a programmer, working with an external expert in server-side applications, and being guided by an educational technology researcher, the mobile App project has been a success. The

regular meetings and subsequent action items provided clarity and transparency. Technological obstacles were overcome as a team; whether they related to code, to server-side variables, to interface design, or to syllabus data.

In addition to the 'search and map' feature, the convenience of the iFUN App enables students to have immediate access to web links such as local information, bus timetables and peer-tutoring schedules, plus twitter feeds of design, technology and science in Japanese and English. Students can quickly access and read about courses and, with personal academic support, student 'learning pathways' can be illustrated thereby empowering students to be motivated to take ownership of their career development.

Moreover, the App has proven cost-effective as it has negated the need to print hundreds of syllabus books (often up to 200 pages each).

Last but not least, due to our 'data-in-the-cloud' approach, syllabus data can now be updated in real-time without the need to specifically update the App itself.

VI. FUTURE RESEARCH

Our future project will adopt a 'wiki' approach where academics and other stakeholders collaborate to continuously update courses and subjects. External stakeholders such as potential employers can also contribute their expectations of university courses (such as particular programming skills, software skills, network diagnostics, effective communication, etc.). By being better informed, course managers and staff can then adjust their style of teaching, the content they teach, and how they assess (in other words, the whole curricula) based upon the evidence displayed. This will help develop new pedagogies within Japanese universities, particularly in science and technology contexts. In addition, an Android version is planned to be available in Chinese and Korean.

REFERENCES

[1] Managing curriculum change. JISC publications. [Online]. Available: http://www.jisc.ac.uk/publications/programmerelated/2009/managingcurriculumchange.aspx

[2] P. A. Towndrow, P.A. and M. Vallance. Making the Right Decisions: Leadership in 1-to-1 Computing in Education. International Journal of Educational Management. Vol. 24, Issue 3. pp. 260-272. 2013.

[3] Titanium Studio. [Online]. Available: URL http://www.appcelerator.com/titanium/titanium-studio/

[4] Wikilinks App on iTunes Store. [Online]. Available: URL https://itunes.apple.com/us/app/wikilinks/id555005650?mt=8

[5] Nodejs. [Online]. Available: http://nodejs.org

[6] TestFlight [Online]. Available: https://www.testflightapp.com

[7] GitHub [Online]. Available: https://github.com/pacsolution/ikaodori

Michael Vallance is an educational technology researcher in the Department of Media Architecture, Future University Hakodate, Japan. His website is at http://www.mvallance.net (E-mail michael@fun.ac.jp)
.
Hiroaki Murayama is a graduate student at Future University Hakodate, Japan (E-mail g2112035@fun.ac.jp)

David Sperling is the Director of Solana Systems Hakodate, Hokkaido, Japan (E-mail: ds@solanasystems.com)