Quanta Education: Year 9/10 Proposal

Frédo Durand, Jingjing Liu, Rob Miller, Una-May O'Reilly,
Chris Terman, Steve Ward, Victor Zue

Overview

The goal of the work proposed below is to bring an engineering perspective to the creation, delivery and assessment of next-generation online teaching and learning platforms. There are opportunities to experiment with new types of content and new interactions, combined in various ways as part of new pedagogies. Using the laboratories of massive, open, online courses (MOOCs), we have, for the first time, the ability to perform education experiments at a meaningful scale, measure the learning outcomes and close the loop by making appropriate changes in the content and delivery. We expect that repeating this engineering cycle will lead to a rapid evolution in how education is delivered and consumed around the world.

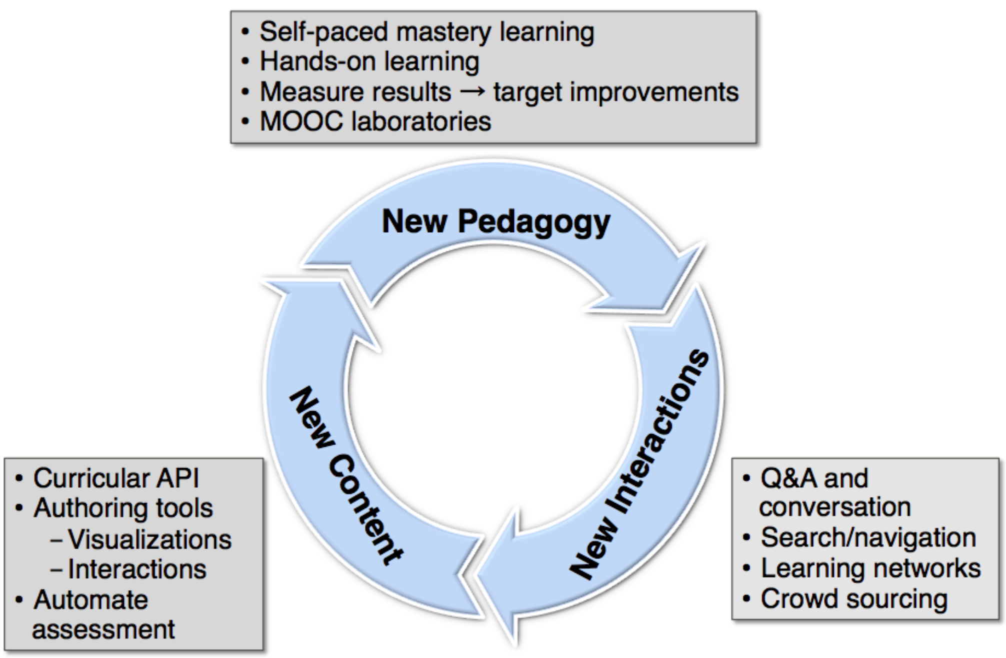

Figure 1. Research Opportunities in Educational Technology

There are two perspectives to be considered. Teachers face interesting challenges in adapting their current materials and techniques to the new realities of MOOCs. And many students will, for the first time, have access to a self-paced, customized educational program that will give them an opportunity to nurture their creativity and collaborate with peers in a learning community that transcends the limitations of their local environment.

Initial forays into creating MOOCs have relied on existing content and traditional lecture-based instructional formats. Research into new authoring tools will let teachers experiment with new content and assessment techniques. Teachers of MOOCs face the challenge of providing timely feedback and supportive learning environments for large cadres of students. Meeting these challenges will require flexible new approaches, *e.g.*, creating feedback using automated techniques for dealing with natural language interactions or by harnessing the power of crowds.
There's the prospect of gathering detailed data on how students interact with the material from which we hope to learn which learning paths yield the best educational results. We propose several investigations focused on these opportunities.

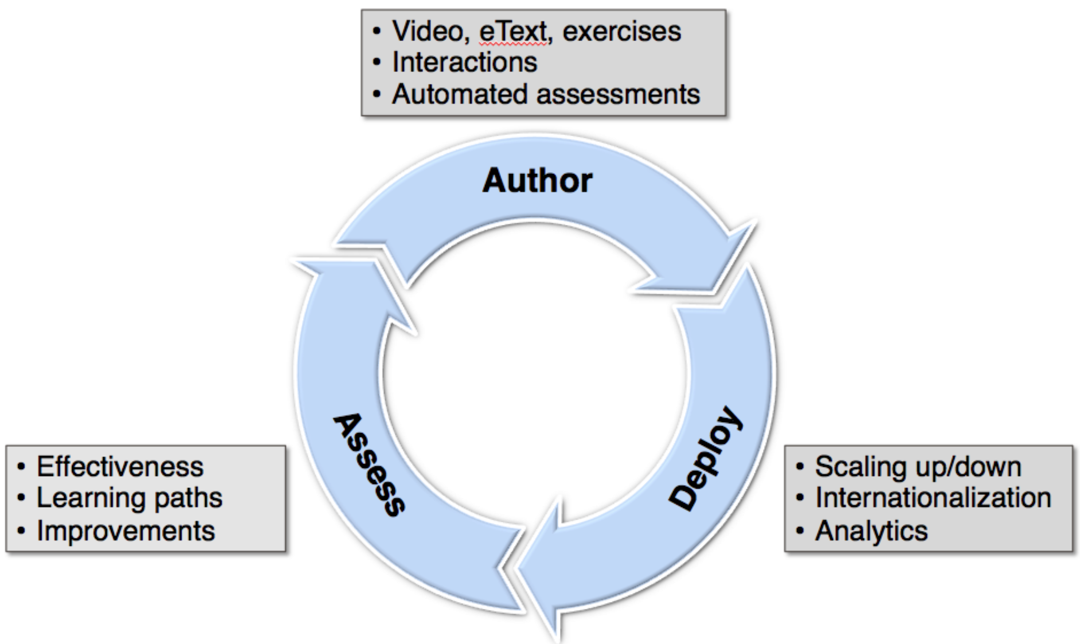

Figure 2. The teaching cycle

Online education provides the opportunity to customize the content and pace to each student's needs. Interactive content allows the student to engage in active learning, which is both motivating and enjoyable, particularly when coupled with peer-to-peer collaboration. Current online collaboration in MOOCs uses forums and other types of networking, but it would be better to incorporate collaboration tools (e.g., a shared virtual venue such as offered by many online games, or simple synchronous communication such as a video chat link). We propose to explore techniques for embedding collaboration into virtual labs. The social aspects of education are extremely important on campus where peer-to-peer learning is perhaps the most effective of all the learning activities. To provide the same opportunity online, we propose to add a mentoring component to courses, asking students who have recently completed a module to help current students with their questions. With careful matching, e.g., by locating mentors who made the same mistake but later resolved their problem, it should be possible to engineer a mutually satisfying experience for both mentor and student. There are two goals of the mentoring activities: first "teaching teaches the teacher," and, second, to use the "massive" nature of MOOCs to provide the large mentoring cadre necessary for timely feedback to current students.

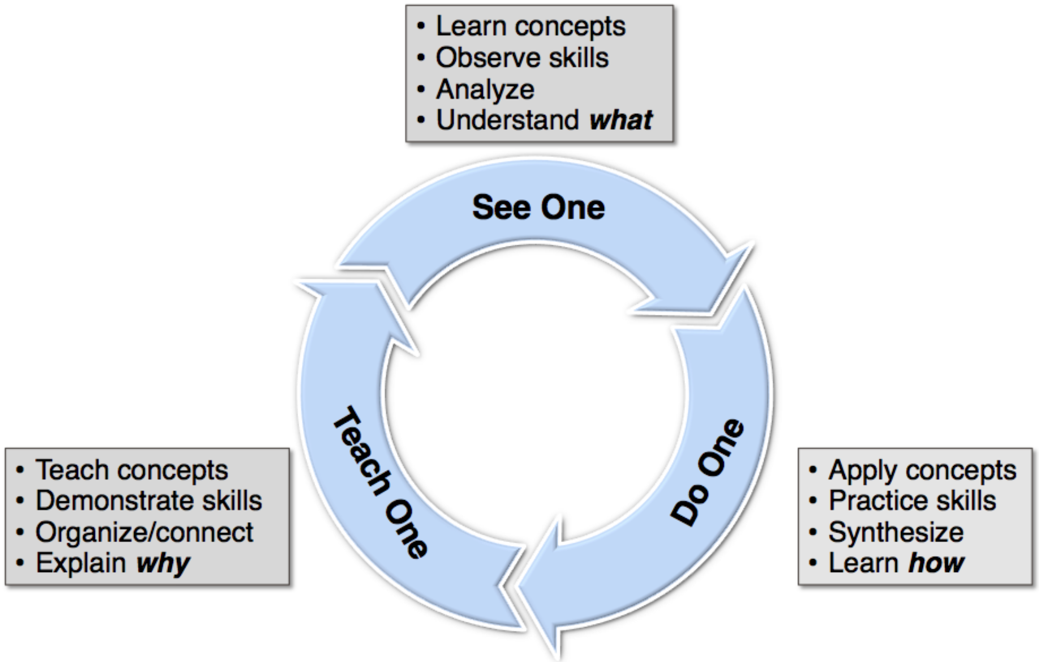

Figure 3. The learning cycle

More detailed descriptions of our proposed investigations appear below.

Bringing Collaboration to Online Hands-on Activities (Chris)

Hands-on activities are very useful in the teaching of science and engineering: performing an experiment or completing a design, then reflecting on the results give students a chance to apply concepts and practice skills via a concrete experience. Hands-on activities help the students engage with the course material while providing motivation and, hopefully, a bit of fun. The increasing power and sophistication of the browser environment makes it possible to build compelling online hands-on activities, especially for recent generations who are experienced denizens of virtual environments.

On-campus students students often perform hands-on activities in environments where it's easy to get over-the-shoulder help from mentors or peers, i.e., where everyone can simultaneously see and manipulate the same artifact or watch someone else do so. The social and technical interchanges during the activity are a key component of their success as a learning activity.

While there are many online communication and collaboration tools supporting both synchronous and asynchronous interactions, these tools are not integrated with the on-line implementations of hands-on activities. The goal of this work is to design, implement and evaluate collaboration frameworks for online hands-on activities, allowing the same sort of social and technical interactions during the activity between remote participants as happen now on campus.

Initially we propose three efforts:

1. Many applications providing an interactive user interface use -- or could easily be transformed to use -- a model-view-controller (MVC) software architecture. Our idea is to create a collaborative MVC where the model is kept synchronized between remote users, so that it can be manipulated through multiple independent views and controllers. For the collaborations to be effective the synchronization needs to happen in near real time. For scalability, it would be best if this involved only peer-to-peer communications, but initially there may a lightweight server required as a central message switch to help address problems arising from users behind firewalls, etc.

2. Provide a simple overlay that would allow remote users to highlight components, add commentary, questions and annotations, etc., without affecting the actual activity, i.e., capabilities similar to "marking up" a PDF document using a PDF viewer. In other words, support for an interaction of the form "Shouldn't this wire be connected here instead of there?" without actually changing the circuit.

3. Augment the various views provided by the online activity to Incorporate existing browser-based A/V chat functionality.

We expect this list to grow as we gain experience from designing, implementing and evaluating these first steps.

Peer Feedback, Mentoring, and Content Creation (Rob)

Both MOOC courses and increasingly-large residential courses suffer from lack of human attention: there aren't enough human teaching staff to provide one-on-one tutoring for individual students. We aim to develop new techniques and systems for peer teaching, that will leverage the crowd of students taking the course, as well as those who have already graduated from the course (the alumni), to give human intelligence and attention.

MOOCs and large residential courses already use discussion forums to get peers to help with question-answering, with significant success. Our approach will broaden the idea of peer teaching in three directions:

* Feedback on other students' work. MOOCs need to move beyond multiple-choice and short-answer questions that can be easily automatically graded, to richer, more creative kinds of student work, including writing, programming, and foreign language use. We will develop systems in which peers to review, comment on, and grade student work. Challenges to overcome include quality control -- making sure the reviewing is high-quality, and reviewers learn to be better reviewers. We will draw from our prior work in crowdsourcing, including techniques like injecting work with known problems into a reviewer's (effectively quizzing them on their reviewing), and having other peers review the reviews.

Mentoring. One of the most effective kinds of learning experiences is the master/apprentice system, doing one-on-one work with an expert mentor overseeing and correcting. We will explore ways to move this online, between peers, in which the peers have been matched so that each is an expert, or more of an expert, in a skill that the other needs to improve. Research challenges here include matching students of appropriate levels, and developing techniques for staff to collect data about the mentoring and monitor the process to produce good learning outcomes.

Content creation. Keeping the course content fresh is a challenge for both MOOCs and large residential courses. Solutions to a problem set or quiz may appear online soon after it is used, reducing its usefulness for learning or assessment because the answers are so easy to get. We propose to develop ways to mine student work to generate quiz or homework exercises. For example, if peer feedback on previous student assignments identifies a common mistake that was explained in a lecture module, then a quiz for that module can be generated from randomly-chosen examples from the previous work.

Knowledge Mining Online Learning Behavioral Data via a Collaborative Technology Platform (Una-May)

Online, blended or MOOC scenarios present an unprecedented opportunity to observe how online technology, content and students interact. We can record and examine the diverse patterns of engagement while relating them to learning outcomes, such as test and problem set performance. We intend to knowledge mine online course behavioral data so that we can help students learn better, teachers teach better and inventors create better-informed online education technology.

One of our goals is to help conduct research on online education behavioral data more easily. Currently, researchers are overwhelmed by multiple streams of low level click stream data in different formats. They need extensive computer science tool knowledge to abstract the data into variables that can inform their models. They build clumsy slow databases that impede the agility of their analysis and their variables are difficult to describe and be used by others - something which is essential to research. We intend to lower this barrier to entry by taking a community oriented perspective. We will set up a technology exchange platform that encourages and supports effective, collaborative online education research. The focal point of the platform will be a standardized, extensible data schema which we will propose and disseminate along with providing examples drawn from MITX courses which use the schema. The platform will enable researchers who have technical expertise to contribute to the community by sharing scripts which extract variables from any database (private or public) where the data conforms to the standardized data schema. Script sharing, supported by a platform that makes the schema accessible, enables returns to scale in visualization methods, variable definition and conditioning. Scripts become the de-facto vehicle enabling cross-institution research collaboration and direct experimental comparison.

Another goal is to invent fundamental new engineering technology for online educational data mining, in this case, machine learning algorithms, which enable important questions which arise in an online context to be addressed. Two example questions are:

High dropout rates have been observed in most MOOCs. Why does this occur? We want to uncover any answers to this question that lie within the student-course behavioral engagement data. The algorithms will analyze how each student navigates through video modules, e-text, and practice questions during each of a course's modules. They will detect behavioral patterns while discerning student cohorts. They will connect the navigation data with information on whether a student stays with the course and earns a certificate or 'drops out'. For example, the data may reveal that the delivery rate of some of the course material is too fast. It would reflect more time being spent reviewing material but no significant attributable improvement in test outcomes by students who have already been discerned as mastering earlier material well and achieving gains relative to how much material they review.

Online courses are stretched to replace the personalized recommendations that come from a teacher observing a student and making suggestions to help them. Given what a student knows now, there is always the ``right'' material to point them to next. How can we provide this help in an online setting? We intend to examine the data and discover what resource usage patterns led to knowledge mastery - as demonstrated, for example, by successful quiz or homework scores. We will identify the level of knowledge a student has, match it to those in the historical data, find among those, the resource usage pattern that led to new knowledge mastery and recommend that.

Flexible content authoring and viewing (Fredo)

We want to dramatically increase the ease and flexibility with which digital educational content is created and consumed. Much current online education relies on videos, which are both costly to produce and rigid to view. Teachers can spend hours per minute of video and need to master complex software and workflow. Furthermore, they often cannot edit content, which makes it hard to maintain and improve material over time, or to augment someone else's material. We will develop new technology to facilitate the authoring and editing of time-based educational content such as video. We will first focus on the handwritten style (e.g. Khan academy), and will then address other styles that show the lecture or the lecturer's hand. Our approach will leverage the special nature of most educational content where, in contrast to regular videos, the spatial dimension (typically corresponding to the black board) is structured and strongly tied to the time dimension. We have identified five important requirements for flexible editing: layout editing, insertion in time, redrawing, time shifting, and flexible synchronization with audio.

We also will develop new online players that support rich student interaction by leveraging the structured nature of our content. Rather than encoding content in a video stream, we will transmit the compact structured vector graphics and will render media in a resolution-independent manner. We will use the strong link between space and time discussed above to enable students to control the player by, for example, clicking on an equation to move to the corresponding explanation. We will also develop methods to support hierarchical content where different levels of explanation are available and can be navigated by the students or automatically chosen by an intelligent tutoring system. We will also facilitate education research by enabling the creation and delivery of different variations of content for A/B testing.

Develop content processing/search capabilities using human language technology (Victor & Jingjing)

MOOC opens the door for many applications that could benefit from human language technologies. For example, we may wish to summarize a lecture in the form of a precis for efficient search, categorize questions raised in course forums so that similar ones can be grouped to establish priorities for answering them, organize the threads of online discussions so that students can better follow them, detect sentiments such as confusion or boredom, so that teachers can modify the delivery of the course material accordingly, provide the opportunity for students to form social communities to promote interactive learning, etc. Many of these will require deep semantic understanding and discourse analysis, grounded in a statistical framework that can grow organically. We plan to use techniques that we have developed in the past to process user-generated content such as reviews and blogs, modified to suit these needs.

Tools for Curricular Design (Steve)

The world of engineering has developed effective techniques for the hierarchical design of complex systems. Intended system behavior is defined in a specification, and alternative system implementations are explored by interconnecting smaller components, each of which is described by its own specification. The logic of a system design is tantamount to a proof that system specs will be obeyed so long as each component obeys its spec.

We propose a framework in which a educational curricula can be designed, documented, and defended using this same proven approach. We view a curriculum as a hierarchy of modules at various levels -- degree programs, full-term subjects, sequences of lectures/laboratory exercises -- each of which is designed to obey a specification, and each of which may be implemented using some combination of smaller educational modules (e.g., the series of lectures and assignments that make up a subject). We represent the design of a curriculum as a hierarchical block diagram whose components are smaller modules interconnected via their specifications, much as a circuit diagram with component specs represents the design of an electronic device.

To facilitate specifications of educational modules, we construct an open-ended universe of shared pedagogic goals. Each goal represents some ability or level of understanding of a specified topic, and is documented using textual descriptions as well as concrete "rubrics" -- each a specific ability, sample test question, or some other measurable milestone of a student's progress. A module specification then includes two disjoint sets of pedagogic goals:

- prerequisite goals, reflecting assumptions about the student's understanding prior to taking that module; and

- outcome goals, documenting the student's understanding upon completion of the module in question.

Each module specification implies a promise made on behalf of that module: that students who satisfy its prerequisite goals and successfully complete the module will satisfy its outcome goals.

Given a set of modules embellished with such specifications, we can assemble curricula in the form of directed graphs whose nodes are modules and their prerequisite and outcome goals. Such a Curricular Map separates modules by intervening goals, i.e. goals that are outcomes of an earlier module as well as prerequisites of a subsequent one.

There may be several available modules with similar specifications, facing the curriculum designer with a design choice. We view this as a step toward the commoditization of educational modules: alternative ways of teaching about caches, for example, will compete in the educational marketplace much as DRAM chips meeting the same spec have competed in the engineering world. Standards, in the form of specifications, are the enabling technology for both examples of commoditization.

We have done some early experimentation with this idea, using MIT EECS subjects as the modules. We propose to develop the approach further, and explore its use on smaller modules amenable to online self-paced mastery learning techniques.

Experimental course modules

We propose to develop some experimental course modules that embody the ideas outlined above. This will allow us to evaluate and improve upon our initial implementations, as well as serving as a concrete demonstration of the work performed. We have some initial ideas of the course modules we would like to try, with final choices to be made as the project progresses:

Beautiful Code: a short course in writing code that is safe from bugs, easy to understand, and ready for change. Intended for people who already have some experience in programming, to give them more software engineering sensibility. Lecture content would cover static and dynamic checking, test-first programming, specifications & exceptions, immutability, don't repeat yourself. This is basically the first 2 weeks of 6.005. Exercises would be small programming assignments, which would be both automatically graded and peer reviewed. Module will be designed so that different programming languages can substituted (Python, Javascript, Java, C#) to support students with different backgrounds. Use of Pentimento will make it easy for others to pick up the lecture content and port it to a new programming language.

User Interface Analysis & Design: a short course in understanding usability and applying it to critique user interfaces and design better ones. Lecture content covers usability, learnability, efficiency, errors, and design principles and patterns for achieving them. Basically the first 2 weeks of 6.813. Exercises would be UI critiques and small redesigns (in HTML/CSS/Javascript), with some automatic grading but mostly peer grading. This is not the usual HCI design course, because it omits all the design process technique (prototyping and evaluation) which are still hard to do well online.

Web Programming Bootcamp: a 2-week, 80-hour intensive course in modern web programming technology, mainly client-side. Topics include HTML, CSS, Javascript, jQuery, node.js, Bootstrap, d3, MVC frameworks. This is a *blended* course, in the sense that it combines heavy use of online resources with residential tutorials, lab help, pair programming, and code review.

Photography 101: This module will teach beginners the basics of picture taking and will cover the basic controls at a user's disposal and will also cover basic aesthetic aspects, with practical tips to make photos better. This module is intended for the broadest audience.

Image formation: This module will teach the technical principles of image formation, starting with pinhole optics, simple lenses, and will go through the standard parameters of a photograph : aperture, shutter speed, ISO, etc. Intended for people with a high-school math background. It will focus on video lectures but will also include several interactive demos. It will roughly correspond to a 2-week class sequence.

Convolution and Fourier analysis in image processing: This module will teach the basics of Fourier analysis motivated by the removal of blur in photographs. It will introduce the Fourier transform as a change of basis that makes convolution diagonal. The module will contain many interactive demos and will encourage students to be active. This module will include programming assignments. Will require basic algebra and calculus.

Computation Structures: one or more modules drawn from 6.004, the EECS Department's introductory course in the engineering of digital systems. Starting with MOS transistors, the course develops of series of building blocks -- logic gates, combinational and sequential circuits, finite-state machines, computers and finally complete systems. Both hardware and software mechanisms are explored through a series of design examples. The organization of course material lends itself to the "see one, do one, teach one" pedagogy we're recommending for MOOCs.