Recasting a Traditional Course into a MOOC by Means of a SPOC

Sébastien Combéfis **Adrien Bibal** **Peter Van Roy**
Computer Science and Engineering Department
ICT, Electronics and Applied Mathematics Institute
Université catholique de Louvain, Louvain-la-Neuve, Belgium
Email: {sebastien.combefis, adrien.bibal, peter.vanroy}@uclouvain.be

Abstract: We give a practical approach to recast an existing, mature traditional university course into a MOOC. The approach has two steps. The first step consists in transforming the existing course into a course with two tracks: a SPOC and a traditional track. This step is done concurrently with teaching the course. The second step, which takes place one semester later, is the opening of the SPOC to the world as a MOOC. We have implemented this approach with the course *LFSAB1402 Informatics 2*, a second-year bachelor university-level programming course taught to all engineering students at Université catholique de Louvain (UCL), that is, approximately 300 students per year. The approach has four advantages. First, it allows to design a SPOC covering only part of the material of the traditional course. A 5 credit (ECTS) one-semester course has almost twice the material of a six-week MOOC with two/three-hour lessons per week. Second, the workload of the transformation is reduced. It can be done incrementally during the teaching of the traditional course. Third, it allows us to gain experience in the world of MOOCs in a relative painless manner. And fourth, since the transformation is a large step, the risk of having problems in the final MOOC is reduced.

Introduction

Online education supported by new technologies is a recent trend that has increasing success. In particular, MOOCs (Cormier, 2008; Downes, 2008), which stands for Massive Open Online Courses, are growing in popularity around the world with companies or associations focused on offering MOOCs with a platform to manage and offer those courses. Examples of such associations include edX, Coursera, Udacity, ITyPA, FutureLearn, and so forth. In particular, UCL joined the edX consortium in early 2013 using the name *LouvainX* and the experience reported in this paper is a consequence of this.

We have taken a traditional, mature course in paradigms of computer programming and we have recast it in a new course containing two tracks: a SPOC (Small Private Online Courses) track and a traditional track. In the new course, the students do one SPOC lesson each week, between two lectures. The purpose of each lecture is twofold: to restructure the material of the SPOC (in the style of a flipped classroom) and to introduce advanced material that is not covered by the SPOC. There is also an exercise session that is held between each two lectures, while the SPOC lesson is ongoing. The purpose of the exercise session is also twofold: first to verify that all students are progressing on the SPOC (again, in the sense of a flipped classroom), and second to provide exercises on the advanced material of the previous lecture.

There are several possible motivations for this organisation. In our case, the primary motivation is to prepare a standalone MOOC. Since the SPOC is local and private, our students play the role of guinea pigs, and their comments and suggestions are instrumental in improving the quality of the SPOC before it is released as a MOOC. The SPOC was given in the Fall semester of the 2013–2014 academic year at UCL, as part of the course LFSAB1402; the MOOC will be given by edX in the Spring semester, starting in February 2014 and called *"Louv1.01x: Paradigms of Computer Programming"*.

A second motivation is to make a first step into the world of teaching using MOOCs. In our case, completely converting the course into a SPOC is not possible nor desirable. It is not possible since the SPOC does not cover all the material of the traditional course. It is not desirable since the step from traditional course to MOOC is quite large, too large in terms of workload for conversion and in terms of risk. We therefore decided to only convert part of the traditional course into a SPOC. The SPOC contains 3 credits (ECTS credits: European Credit Transfer System, a harmonized unit of course size at European level) of material, while the course contains 5 credits.

There are three challenges in creating the SPOC. Firstly, since the MOOC must serve on-site students for the already existing course, this means that its integration as a SPOC will greatly influence its structure and organisation. Secondly, the main objective of the course being to teach programming, assessing the students in the SPOC must be performed by checking their ability to produce code, which is a big technical challenge. Finally, the last challenge is social, requiring several teams and training staff to collaborate around a common project, each bringing their own skills.

The remainder of the paper is structured as follows. The first section is about the structure and the timeline of the remastered course, which integrates the SPOC. The second section is about technological and social aspects that have been put together to serve the recasting of the traditional course. Finally, the last section presents a first evaluation of the remastered course and of the workload of the training team and of the students.

Structure and Timeline of the Course

The developed MOOC is based on an existing traditional course that has been taught since 2005, this year being the 9th edition (Van Roy, 2011). The course is taught to all second-year bachelor students in engineering (not restricted to computer engineering students) and also in computer science. Around 300 students every year take the course. As previously mentioned, the idea is to integrate the MOOC with the existing course. The first challenge was therefore to choose a way to integrate the MOOC with the existing course, while keeping the MOOC as an individual, self-contained, and relevant course. The solution that was chosen is to split the material of the course in two tracks: one that is supported by a SPOC, a private version of the MOOC, and the other one that continues to be taught with a traditional course. The SPOC track contains slightly more material than the traditional track. The SPOC track lays the basis whereas the traditional track comes with more examples to illustrate the material covered by the first track and also brings advanced concepts. The course is split into cycles that all follow the same pattern illustrated by Figure 1. One cycle corresponds to one week, and the course lasts twelve weeks. Moreover, to each cycle corresponds one SPOC lesson.

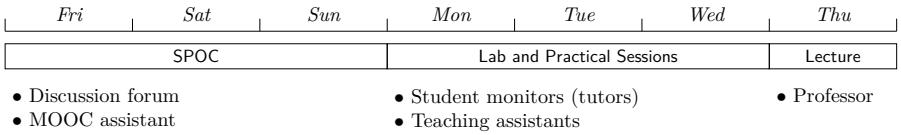

Figure 1. Organisation of a classical week for the remastered course.

The first activity of the cycle is the SPOC part. Firstly, the students have to follow the SPOC track on their own. The SPOC track is implemented with the edX Studio Platform; it consists of videos presenting the theory complemented by documents and exercises of various kinds. Videos are kept as short as possible and last between 5 and 10 minutes. Exercises are interleaved with videos, that is, a set of exercises is proposed after each video or small group of videos. Two kinds of exercises are proposed to the students: *classical* and *coding* exercises. Classical exercises consist of multiple-choice questions and questions for which the answer is a word or a few words. Coding exercises ask students to write code fragments that are then assessed by an automatic tool to provide an intelligent feedback to the student. In case of issues, students can use online discussion forums or send requests to a MOOC assistant.

After having worked on the SPOC part, students have to attend a two-hour exercise session. The first hour is dedicated to the SPOC track. Students are supervised by student monitors and teaching assistants whose goal is to animate the session following an active learning approach inspired from problem-based learning. The goal of this first part of the session is to ensure that all the students have understood the SPOC exercises and related theoretical aspects. During the second hour, students receive supplemental on-paper exercises covering the advanced aspects of the traditional course track.

Finally, the last activity of the weekly cycle is a two-hour lecture given by the professor. The lecture is also split in two parts. The first part acts as a restructuring lecture. During the first part, the SPOC lesson is restructured and situated with respect to upcoming lessons. The second part is dedicated to the traditional course track and presents advanced concepts and more examples on the material covered by the SPOC track. Due to the timing of the organisation in two tracks, there is a shift between the two tracks. As shown in Figure 2, the advanced exercises of the practical sessions of a week are related to the material covered by the lecture of the previous week.

The first week of the semester, before the first cycle, one introduction lecture is given to students. The goal of that first lecture is double. Firstly, it introduces the logistic and organisational aspects to the students, and explains the precise structure of the course, presenting the whole training team. Secondly, it introduces briefly the SPOC of the first cycle.

	SPOC	Practical Session	Lecture
SPOC Track	*Video + exercises* (i)	*Feedback* (i)	*Restructuring* (i)
Traditional Course Track		*Advanced exercises* $(i-1)$	*Advanced concepts* (i)

Figure 2. Decomposition of the course into SPOC and traditional course tracks.

The MOOC is integrated into the course following the flipped classroom pedagogical model (Lage, Platt, & Treglia, 2000). Indeed, students discover the new materials and are training themselves while being at

home. Then they interact with the training team when they are on-site. An important property of this model for us is that the restructuring part of the lecture is given in French. It is important since the native language of the majority of the students is French, and because they are second-year bachelor students and not used to have courses in English since it is not a prerequisite. The MOOC is in fact used as a SPOC for the on-site course during this semester, just as it has been done at Harvard (Coughlan, 2013). The public MOOC will be open to the world next Spring. Since it will only last six weeks, online students will have to work on two cycles a week.

The remastered course is evaluated exactly as the traditional course. It is impossible to evaluate SPOC performance with the level of identify verification required for traditional courses leading to a diploma. Usually, MOOCs that offer this kind of evaluation offer proctored examinations in addition to the online course. In our hybrid course, therefore, we give only a small weight to the SPOC evaluation. We keep the full evaluation structure of the traditional course: a mandatory project done during the semester, a written midterm exam (one hour), and a written final exam (three hours). Both exams require writing program code. The final score combines the project (*1/4*) and the final exam (*3/4*), and the midterm is used only to increase the score using a weighting (*max (e, 2e/3 + m/3)* where *e* = project plus final exam and *m* = midterm). The SPOC intervenes only through an incentivisation scheme: a bonus is added to the final exam based on participation in the SPOC. We do not claim any particular properties for this evaluation scheme, except that it works well for us and is accepted by the engineering school. Of course, another question that is directly raised is how students that will follow the MOOC during the next semester will be evaluated. The idea is to propose a mid-term and final evaluation, composed of classical and coding exercises. This evaluation will already be included into the SPOC during the Fall, and will serve as review exercises for the material covered by the SPOC track.

It is important to incentivise the students to do the weekly SPOC lesson before the lecture. If we simply request their participation, then experience shows the majority of students will not do it. University students have a busy schedule with many courses, labs, and projects, and they also have a busy social life. To incentivise students to do the weekly SPOC lesson with its Pythia exercises, we keep statistics on who has participated and make them public. Completion of all exercises of all 12 lessons earns a *+2* bonus on the final exam score (out of 20). Zero participation earns a *-2* penalty, and intermediate levels give a proportional bonus. This scheme is based on a similar bonus scheme we have devised in the traditional course to encourage weekly programming exercises. Experience over the five years that this scheme has been used shows that it is highly successful in incentivising our students to do weekly assignments (your mileage may vary: the usefulness of such a scheme depends on the student culture at your institution). Doing regular exercises every week improves the students' understanding of the material significantly.

Technological and Social Integration

In order to reach all the objectives of the remastered course, both technological and social aspects have been taken into account from the beginning. As previously mentioned, developing a MOOC for a course where the students have to produce code requires the ability to assess the produced code. The technological aspect is therefore to provide relevant exercises that can be assessed automatically and for which feedback can be given to the students. The automatic part is important for the scalability of the MOOC and the feedback part is important to support student learning. We use the Pythia platform (Combéfis & Le Clément de Saint-Marcq, 2012) to satisfy both requirements. Pythia is an online platform that can automatically grade programs while providing intelligent and relevant feedback. The platform has the ability to execute programs safely, so malicious code cannot destroy the platform. Creating Pythia tasks is somewhat time-consuming, in particular to identify *a priori* what mistakes the students will make. Identifying common mistakes is necessary to provide dedicated feedback messages that will drive students toward the right answer. Student use of Pythia during the SPOC stage allowed us to improve the detection of common mistakes.

The social aspect is also very important for the success of the remastered course and the MOOC. Several domains of expertise have been combined in order to develop the MOOC and the remastered course. First of all, the professor is in charge of creating the videos that are used for the MOOC. This is done without the help of any audiovisual service, with just a personal computer, a Wacom Cintiq graphic tablet, a webcam, a good-quality microphone and Camtasia software. Using the content of our mature course, we find it takes about one full day of work by the professor to produce 45 minutes of video. In addition to the professor, a part-time MOOC assistant is in charge of developing the exercises (classical and coding ones) for every SPOC lesson. The MOOC assistant also animates a discussion forum about the exercises, in order to clarify and improve the Pythia tasks according to the mistakes made by students. Finally, a part-time research assistant develops the Pythia platform, incorporating improvements and to get a stable and scalable version for the MOOC of the next semester. Table 1 summarises all the course staff with their roles (for a class of 300 students). In addition to this staff, the university also offers the part-time assistance of one specialist in pedagogy and e-learning. Finally, a steering committee coordinates the creation of MOOCs at the university level (since our MOOC is part of four MOOCs being developed during the fall semester for release in 2014).

Table 1: Training staff for the two-track remastered course and their roles.

Staff	#	Role
Professor	1	MOOC video creation, lectures
MOOC assistant	0.5	MOOC exercise creation, discussion forum animation
Research assistant	0.5	Pythia platform development
Teaching assistant	4	Practical session management and supervision
Student monitor (tutor)	11	Practical session supervision

The professor, the MOOC assistant, and the research assistant collaborate closely and regularly provide each other with feedback about the current situation. Indeed, the course is being remastered while simultaneously being taught to the students. We find that this situation is workable if the course preparation is at least two weeks ahead of the students. Moreover, the teaching assistants (who are already used to the traditional course) and also the student monitors (who have previously taken the traditional course) need at least one week lead time to prepare their sessions. Even if the technical content of the remastered course is the same as last year's traditional course, it is important that all the course staff agrees on the remastered course and appears as a unified team to the students.

Evaluation of the remastered course and workload

The remastered course and SPOC were completed by the end of the Fall semester. We are still in the process of evaluating the remastered course and the SPOC, and this evaluation will not be complete until the MOOC is complete. We have made a quick poll among the students to evaluate their workload for the SPOC exercises. The question that was asked to the students is the mean time required for solving one classical exercise and for solving one coding exercise. Table 2 shows the results of the poll. Not all students have replied, not have replied to both questions.

Table 2: Workload to solve one exercise.

	Classical exercise	Coding exercise
Less than 5 minutes	36	10
5 minutes	66	46
10 minutes	4	29
15 minutes	0	1
More than 15 minutes	1	25

On average, coding exercises do not take much more time for students than classical exercises. This observation can be counter-intuitive at first since coding exercises are more complex for students than multiple-choice questions. One possible explanation is that since programming is the focus of the course, the students are considering that it is indeed an objective to be able to program and therefore students are more careful during the videos and more engaged to solve the coding exercises. Another possible explanation is that their immersion makes them underestimate the time they think they are spending on the coding exercises. A final observation is that 25% of the coding exercises take significantly more time, when students "get stuck". This is an unacceptably high percentage and we are working on improving the exercises and the Pythia feedback to reduce this percentage. Our Pythia statistics tell us how many attempts students make for each exercise and the Pythia platform stores all the attempts, which allows us to focus on the troublesome exercises.

Conclusion

Building a MOOC, even if based on an already existing and mature course, is not an easy task at all. It requires many resources in staff and time. However, as we explain in this paper, creating a new MOOC from an existing course can be taken as an opportunity for a recasting of the course. The possibility to create a SPOC for on-site students, either to replace the course or to integrate it with the existing course, as proposed in this paper, is very useful to test the future work, and to gain experience with limited risk.

As of end December 2013, we have created and given the full remastered course with the SPOC track to all second-year engineering students at UCL (300 students). The SPOC has significantly improved during the semester through our growing experience and feedback from students and teaching assistants. For the MOOC that will open in February 2014, we will make a second pass through the SPOC to incorporate these improvements uniformly throughout the whole course. We also intend to make a comparison of students performance with and without the SPOC.

References

Combéfis, S., & le Clément de Saint-Marcq, V. (2012). Teaching programming and algorithm design with Pythia, a web-based learning platform. *Olympiads in Informatics, 6*, 31-43.

Cormier, D. (2008). The CCK08 MOOC – Connectivism course, 1/4 way, retrieved September 24, 2013 from http://davecormier.com/edblog/2008/10/02/the-cck08-mooc-connectivism-course-14-way/

Coughlan, S. (2013). Harvard plans to boldly go with 'Spocs', retrieved September 24, 2013 from http://www.bbc.co.uk/news/business-24166247

Downes, S. (2008). CCK08 – The distributed course, retrieved September 24, 2013, from https://sites.google.com/site/themoocguide/3-cck08---the-distributed-course

Lage, M., Platt, G. J., & Treglia M. (2000). Inverting the classroom: a gateway to creating an inclusive learning environment. *Journal of Economics Education, 31 (1)*, 30-43.

Van Roy, P. (2011). The CTM approach for teaching and learning programming. *Horizons in Computer Science Research, 2*, 101-126.