

Experiences in Using a MOOC to Support Teacher Professional Development of a New AP Course in Computer Science

Md Mokter Hossain

Jeff Gray

The University of Alabama

United States

mokter@gmail.com

gray@cs.ua.edu

Abstract: This paper presents our experience in designing and delivering a free multi-week Massive Open Online Course (MOOC) with Google Course Builder (GCB), which was deployed on the Google App Engine (GAE) cloud server. The course was offered mainly to high school teachers in the US who were interested in learning about the curriculum and pedagogical strategies for teaching a new Advanced Placement (AP) course that is being created by the College Board. The new course, called Computer Science Principles (CSP), is designed to cover topics about computer science in a manner that is appealing to a broad range of students. More than one thousand CS teachers from 45 US states and 6 countries participated in this online workshop. Our experience suggests that the freely available features available in Course Builder offer many advantages for designing and delivering MOOCs, yet there are some challenges that we faced in our initial adoption. This paper describes our experience as first-time MOOC developers and users of Course Builder, and offers advice for others who may be contemplating their own MOOC.

Keywords: App Engine, Cloud Computing, Course Builder, MOOC, Open Online Course

Introduction

A Massive Open Online Course (MOOC) is an online course that aims to deliver educational services to a large number of participants, often with free, open, and unlimited participation. In addition to traditional web-based course materials (e.g., text, audio, video, and problem sets), a MOOC provides interactive user forums that help build a collaborative virtual online community among teachers, students, researchers, and educators. MOOCs are a recent development in online education that began to emerge in 2012 [5]. According to *The New York Times*, 2012 became “the year of the MOOC” as several well-financed providers emerged in association with top universities (e.g., Coursera, Udacity, and edX) [9]. Within this short time many leading universities, for-and non-profit organizations and prominent educators, have been working on MOOC design and implementation. According to recent statistics on MOOCs [6], 26.5% of US universities; 16.5% of Indian universities; 16.2% of Chinese universities; 3.9% of Canadian, Australian, and UK universities provide MOOCs. Much debate has ensued about whether the many promises of MOOCs can be realized. In our own work, we have explored the potential of using MOOCs to offer teacher professional development (PD) at a national level. The intention of participants to enroll in a PD-focused MOOC may be different than a traditional MOOC. This paper summarizes our initial experience for those who may be interested in developing their own MOOC, particularly within the context of the Course Builder platform.

Background Study

MOOCs are made possible by a successful integration of Web 2.0 technologies and Cloud computing to provide teaching-learning services in an online manner. Web 2.0 refers to the second generation of Internet services that provides Internet users many interactive, dynamic, and collaborative services. Web 2.0 allows users to access and customize information, which has made web design available for users to customize and share their resources freely [8].

Over the past few years, many schools, colleges, universities, as well as profit and not-profit organizations and individual researchers, teachers, and educators have been developing and delivering Web 2.0-assisted online education services. Using the interactive features of Web 2.0 technology, Online Course Management Systems (OCMSs) such as

Angel, Blackboard Learn, eCollege, and WebCT have been developed to provide online and blended teaching-learning systems. These and many other Web 2.0-based OCMSs have recently gained popularity and have been used widely worldwide. However, the major limitations of the OCMS-based courses is their relative high cost. Such courses are provided only through the use of supportive tools and technologies that require additional cost, installation, and support in both school and home computers that may have different configurations, operating systems, and security setups [11]. Moreover, students' and instructors' access to a WebCT course is limited for a certain time only. After the course or semester is over students do not have access to the course content. Most notably, designing courses on an OCMS requires information technology savvy instructors that are sometimes unavailable or unaffordable by the schools in rural districts or in developing and underdeveloped countries [4].

The problems and challenges of traditional and OCMS-based online courses led to the emergence of MOOCs with unlimited and open accessibility. While earlier MOOCs mainly emphasized open access features, such as social and cultural connections and open licensing of content, newer MOOCs use closed licenses for their course materials but maintain free access for students [1, 2]. As MOOCs facilitate a relatively free and unlimited accessibility, they can provide various new dimensions to meet the diversified needs across international and cultural boundaries. MOOCs allow instructors and designers to manage and control online courses easier than traditional static online teaching-learning systems. The online discussion forums provide the means to decrease many of the limitations of the earlier online systems and increase learners' problem solving by following a similar method suggested by other learners or instructors.

In July 2012, Google inaugurated Course Builder as an experimental project for delivering online courses [7]. Course Builder is an online course management tool that allows instructional designers to create interactive teaching materials for web-based training courses. Google Course Builder (GCB) is an open source online platform that deploys course materials on Google App Engine (GAE). The App Engine is a cloud computing platform that provides services for developing and hosting web applications in Google-managed data centers [12]. App engine was first released in April 2008 and came out of preview in September 2011.

App Engine offers automatic scaling for web applications, which means that as the number of user requests increases for an application, App Engine automatically allocates more resources for the web application to handle its additional demand [10]. The App Engine infrastructure handles all of the distribution, replication and load balancing of the entire course content and students' records [3]. Use of App Engine services is free up to a certain level of consumed resources, 5 GB for Cloud Storage Standard Storage; 1 GB for Datastore Stored Data; 1GB Outgoing Bandwidth, 28 Frontend Instance Hours; 9 Backend Instance Hours. Additional storage, bandwidth, or instance hours required by the application are available to the course developer for a fee.

Course Builder was developed to help teachers and educators create online courses and deliver the course to a large range of online class sizes, based on the underlying capability of App Engine as a cloud server. Google itself launched several MOOCs, such as Power Searching with Google V1 in July 2012; followed by Power Searching with Google V2 in September where about 154,000 and 124,000 students registered in these courses, respectively. According to a report on a Google Research Blog, the average student satisfaction rate in these MOOCs was more than 4.44 on a 5.0 scale [13]. This suggests that Google technologies can help bring education to a global audience. Consequently, Google made the Course Builder technology open and accessible for educators and researchers so that they can develop their own open courses [7].

The professional development MOOC that we offered to high school teachers is another example of using Course Builder for the design and delivery of a MOOC offered for high school teachers interested in offering the new College Board Computer Science Principles (CSP) course in their classrooms. Our MOOC is still open for participation and available at: <https://csp-cs4hs.appspot.com>

This paper provides an introduction to MOOCs and briefly discusses the methodology and procedures we used in conducting our experiments that document our experiences. The paper mainly focuses on the problems faced during the development and administration phases of our MOOC and summarizes the actions taken to resolve the problems. We offer recommendations on how to develop a MOOC for the first time, as well as specific discussion on the Course Builder platform, which is attractive to many as a free and open solution.

Procedure

The aim of this study was to design and deliver a MOOC using Course Builder that focused on Computer Science Principles. Before implementing the real course, we deployed a test course to experiment and configure the desired features that we wanted to provide. Permission was sought from our university Institutional Review Board (IRB) for conducting pre- and post-surveys with the participants.

Research Questions

The questions that we sought to address in our experience are as follows:

Research question 1: What are the advantages of using Course Builder in designing and implementing a MOOC over a traditional online course management system?

Research question 2: Are there any requirements or open challenges for joining and completing an open online course using Course Builder?

Research question 3: What are the requirements for teachers and researchers in designing and implementing a MOOC on Course Builder?

Research question 4: What are the main challenges in designing and administrating a MOOC and what strategies exist for handling these challenges?

Findings and Discussion

Advantages of Using Course Builder

There are a number of advanced features in Course Builder for designing and implementing MOOCs over traditional OCMSs. For instance, Course Builder provides a great opportunity for teachers and educators for designing and implementing a MOOC without having a deep programming background. Use of Course Builder is relatively cost effective compared to traditional online course management tools such as Angel, Blackboard Learn, WebCT, and/or many Web-based applications. Course Builder provides an open, free, and 24/7 accessibility for the MOOC participants. Unlike blogs and many other Web 2.0 tools, Course Builder is more flexible in using formatted texts, links, images, videos, activities, Google Docs, and many other static and dynamic features in a lesson. Adding and editing a New Course, Unit, Lesson, Activity, Assessment and/or link is facilitated through a web-based dashboard. A Course Builder course can include any number of Units, where any number of lessons and activities can be included in a Unit. Thus, Course Builder is appropriate for designing and implementing any kind of course that may have different kinds of lessons, activities, and units. Course Builder can be used for designing and delivering any kind of online or blended class.

CS Principles MOOC Participant Information

In order to join and complete an open online course, a participant does not need to have any programming knowledge unless that is required for the course itself. However, each participant must have a Gmail address or at least a Google credential email address for registering in a Course Builder course.

There are over 1,050 participants enrolled in our CS Principles MOOC. In order to know the demographic information of the participants, we included an anonymous Pre-survey in the course. After registering into the class, participants were asked to complete the Pre-survey. Five-hundred and eighty-eight participants completed the Pre-survey. These were from 46 US states and from some other countries including, Australia, Bangladesh, Canada, China, India, Spain, France, Germany, Nigeria, Paraguay, Singapore, and South Korea. Among the participants, about 73% were public high school teachers, 20% were private school teachers, and the remaining were charter school teachers and after school program affiliates.

Table – 1. Participants' teaching level by school type cross tabulation

School Type → ----- Teaching Level ↓		Public School	Private School	Charter School	After school program	null	Total
Elementary	0	4	3	0	1	0	8 (1.4%)
Middle School	0	33	18	0	3	2	56 (9.5%)
High School	0	341	77	12	0	3	433 (73.6%)
Post-Secondary	1	29	9	1	0	0	40 (6.8%)
Currently not Teaching	0	19	6	1	4	11	41 (7%)
Null	0	4	2	0	0	4	10 (1.7%)
Total	1	430 (73.1%)	115 (19.6%)	14 (2.4%)	8 (1.4%)	20 (3.4%)	588

Among the high school teachers more than 37.4% were currently teaching a CSP course. Moreover, about 41.6% were not sure but might be teaching a CSP course in the near future. Among the public and private school teachers, these rates were about 36.8% and 39.3%; and 38.3% and 44.4%, respectively. For more detail, please see Table -1 and Table-2.

Due to several requests by some participants, the duration of our MOOC was extended through mid-Fall. In the course schedule, there is a post-survey that we have not yet completed since the course is ongoing.

Table – 2. Participants’ teaching level and school type by Plan of Teaching a CSP course cross tabulation

		Yes	No	Not Sure	null	Total
Teaching Level	Elementary	2	1	5	0	8
	Middle School	25	10	21	0	56
	High School	162 (37.4%)	91 (21.1%)	180 (41.6%)	0	433 (100%)
	Post-Secondary	12	13	14	1	40
	Currently not Teaching	4	22	15	0	41
	null	3	3	3	1	10
Total		208	140	238	2	588
School Type	Public School	158 (36.8%)	102 (23.8%)	169 (39.3%)	1	430 (100%)
	Private School	44 (38.3%)	20 (17.4%)	51 (44.4%)	0	115 (100%)
	Charter School	4	2	8	0	14
	After school program	1	3	4	0	8
	null	1	12	6	1	20
	Total	208	140	238	2	588

Suggestions for the MOOC Developers

The only external requirement for accessing the resources on Course Builder is a Gmail address or at least a Google credential email address for creating an App on the Google App Engine. Based on our initial experience while implementing our MOOC, we realized that some level of programming, particularly in Python, would be helpful for anyone who wants to use the full power of Course Builder to design a new course. In this section, we discuss some of the potential challenges that may be faced when using Course Builder for the first time, and we suggest strategies for addressing these challenges.

Accessing Student Data

In this section, we briefly summarize strategies that we adopted when using Course Builder. *First*, the Course Builder Get Started wiki page <http://code.google.com/p/course-builder/wiki/GetStarted> discusses the basic need and steps for getting started with Course Builder. The first page (title: Download the Software) describes how to download Python, Google App Engine, and Course Builder. The second page (title: Run Course Builder) describes how to run Course Builder for the first time. The third page (title: Explore Course Builder) describes what participating students see when they take the course. Finally, the fourth page (title: Deploy to Google App Engine) describes how to deploy Course Builder to Google App Engine so that a new course can be instantiated. In this page there is a link¹ to three short YouTube videos describing all of the above steps - from the installation download to the creation of a new course. The third video shows how to deploy Course Builder to Google App Engine through using the `appcfg.py` with command line instructions. However, this step can also be accomplished more efficiently by selecting the desired App and then clicking on the Deploy button in the Launcher.

Second, we found that Course Builder has some kind of Color incompatibility with different browsers. For instance, we were trying to replace Course Builder’s Navigation Bar’s default gray color by our favorite Crimson color (#990000). After editing the corresponding CSS program codes, we did not see the effect in one of our desktop computers. We did not notice that the color change was effective in our Google Chrome browser, but not on Internet Explorer and Firefox browsers. Later, we noticed that some of the gradient styles defined in the Course Builder CSS were not supported by all browsers. These issues were more cosmetic and did not affect the functionality of our course

Third, the App Engine can Backup selected entries; restore the backed up entries to the same or another App; and delete the selected entries. However, it has no such option to download the any kind of entries directly from the course. This might be a challenge for many MOOC course developers during their first use. In order to download Course Data from the App Engine cloud server, Course Builder documentation recommends to configure file paths by creating a Python program file named `gcbpaths.py`, which is saved into the `\tools\etl` folder, and then executed from a command line statement with the following format:

¹ <http://www.youtube.com/watch?v=OtVd1komyGY&list=PLbTy14-ZMIDJGS1XLdYj3Nzrhfd6EDWJC>

```
python tools/etl/gcbpaths.py download course <course_url_prefix> <app_name> <app_server_name> --archive_path <archive_path>
```

After a Google Hangout conference with a Course Builder specialist from Google, we learned that the instructions stated for downloading course and student information from the App Engine's cloud server are compatible with Mac and Linux operating systems and do not work with Windows. We discovered a workaround solution posted on the Google App Engine developers list at: https://developers.google.com/appengine/docs/python/tools/uploadingdata#Python_Using_automatic_configuration

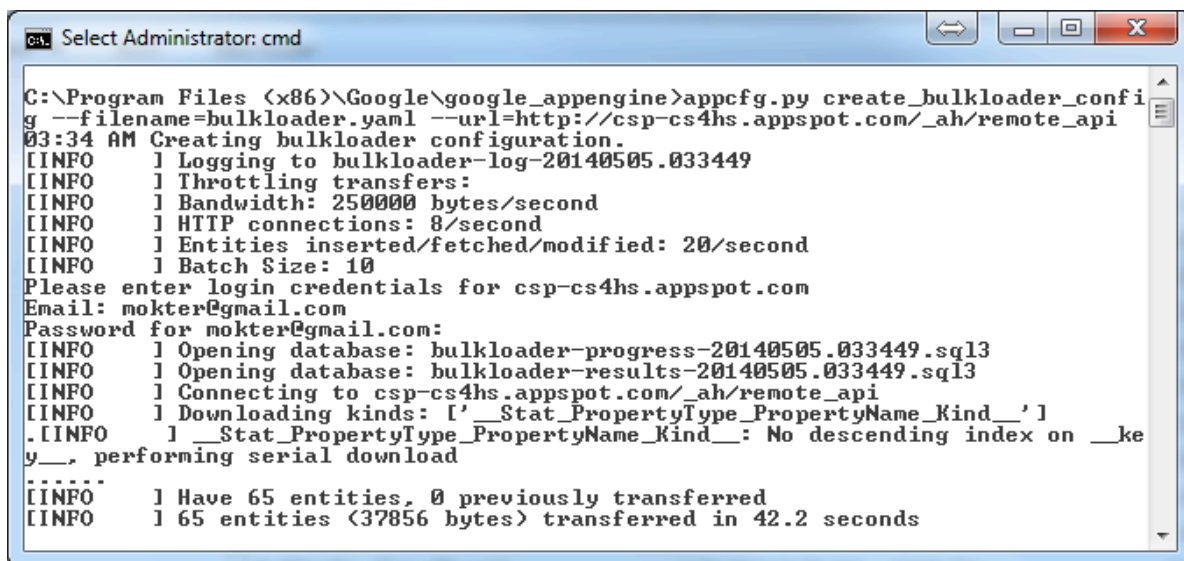
Following the instruction stated in the above page, we tried with the following commands:

```
CD C:\Program Files (x86)\Google\google_appengine
appcfg.py create_bulkloader_config --filename=bulkloader.yaml --url=http://csp-
cs4hs.appspot.com/_ah/remote_api
```

This was according to their recommended format:

```
appcfg.py create_bulkloader_config --filename=bulkloader.yaml --
url=http://<your_app_url>.appspot.com/_ah/remote_api
```

It should be noted that this appcfg.py file is found in the *C:\Program Files (x86)\Google\google_appengine* directory for our PC with 64-bit Windows 7 OS. For a 32-bit version of Windows 7 it should be found in the *C:\Program Files\Google\google_appengine* directory.



```

C:\Program Files (x86)\Google\google_appengine>appcfg.py create_bulkloader_config --filename=bulkloader.yaml --url=http://csp-cs4hs.appspot.com/_ah/remote_api
03:34 AM Creating bulkloader configuration.
[INFO] | Logging to bulkloader-log-20140505.033449
[INFO] | Throttling transfers:
[INFO] | Bandwidth: 250000 bytes/second
[INFO] | HTTP connections: 8/second
[INFO] | Entities inserted/fetched/modified: 20/second
[INFO] | Batch Size: 10
Please enter login credentials for csp-cs4hs.appspot.com
Email: mokter@gmail.com
Password for mokter@gmail.com:
[INFO] | Opening database: bulkloader-progress-20140505.033449.sql3
[INFO] | Opening database: bulkloader-results-20140505.033449.sql3
[INFO] | Connecting to csp-cs4hs.appspot.com/_ah/remote_api
[INFO] | Downloading kinds: ['__Stat_PropertyType_PropertyName_Kind__']
[INFO] | __Stat_PropertyType_PropertyName_Kind__: No descending index on __key___. performing serial download
[INFO] | Have 65 entities. 0 previously transferred
[INFO] | 65 entities (37856 bytes) transferred in 42.2 seconds
```

Figure 1. Successful download of bulkloader.yaml using appcfg.py in another desktop Windows 7

The above command creates an auto generated *bulkloader.yaml* file that comes without any connector type, which needs to be edited as:

```
connector: # TODO: Choose a connector here: csv, simplexml, etc...
```

using our desired connector type “csv” as shown below.

```
connector: csv # TODO: Choose a connector here: csv, simplexml, etc...
```

There were about 15 places where this replacement was needed. The connector_options were kept unchanged. Additionally, there were another set of changes that were needed, as follows:

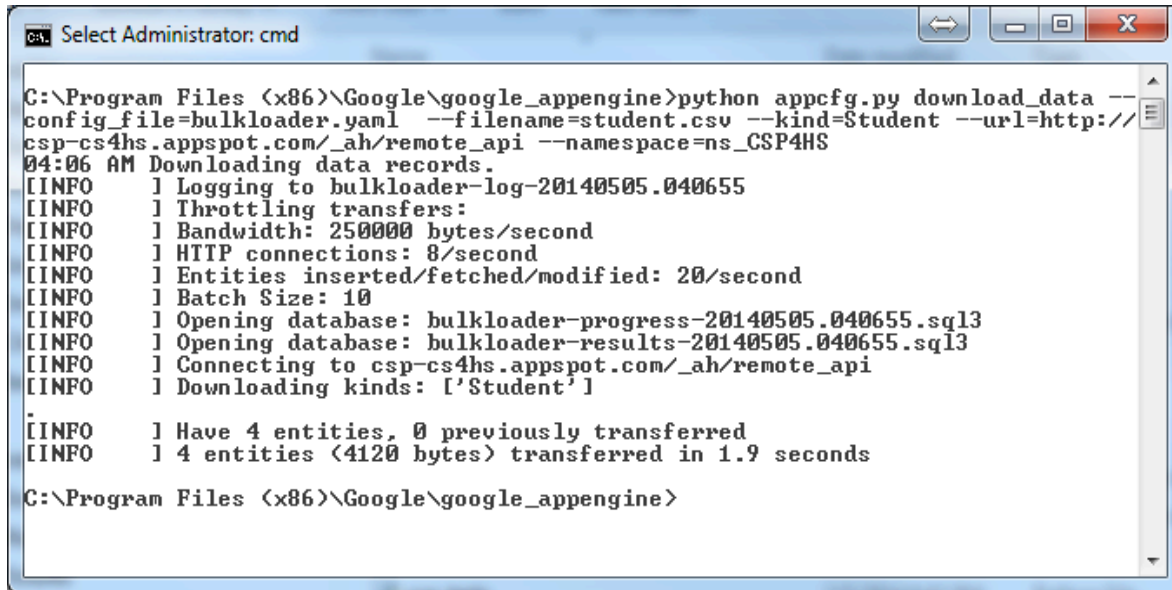
```
import_transform: transform.create_foreign_key('TODO: fill in Kind name')
```

Where it was asked “TODO: fill in Kind name”. We edited it as follows:

```
import_transform: transform.create_foreign_key('Reviewee')
```

Then we were able to download the Student data from the Google App Engine.

```
python appcfg.py download_data --config_file=bulkloader.yaml --filename=student.csv --kind=Student --url=http://csp-cs4hs.appspot.com/_ah/remote_api --namespace=ns_CSP4HS
```



```

C:\Program Files (x86)\Google\google_appengine>python appcfg.py download_data --
config_file=bulkloader.yaml --filename=student.csv --kind=Student --url=http://
csp-cs4hs.appspot.com/_ah/remote_api --namespace=ns_CSP4HS
04:06 AM Downloading data records.
[INFO]   | Logging to bulkloader-log-20140505.040655
[INFO]   | Throttling transfers:
[INFO]   | Bandwidth: 250000 bytes/second
[INFO]   | HTTP connections: 8/second
[INFO]   | Entities inserted/fetched/modified: 20/second
[INFO]   | Batch Size: 10
[INFO]   | Opening database: bulkloader-progress-20140505.040655.sql3
[INFO]   | Opening database: bulkloader-results-20140505.040655.sql3
[INFO]   | Connecting to csp-cs4hs.appspot.com/_ah/remote_api
[INFO]   | Downloading kinds: ['Student']

[INFO]   | Have 4 entities, 0 previously transferred
[INFO]   | 4 entities (4120 bytes) transferred in 1.9 seconds

C:\Program Files (x86)\Google\google_appengine>
```

Figure 2. Successful download of student.csv data using the appcfg.py file with customized bulkloader.yaml

The processes of downloading student data from the App Engine cloud data storage took much investigation because the same process did not work on our other computer where we usually did most of the course development activities. Later, we came to know that these commands should be executed on a Mac and/or Linux OS. As Course Builder evolves, capabilities and features will emerge to address these initial accidental complexities.

Implications of the Study

The results of this study have possible implications for researchers, educators, and students regarding the use of MOOCs for synchronous and asynchronous learning. The observations of this study might provide a great opportunity for teachers and educators who do not have enough programming background, but wish to design and implement an open online class for their students and fellow researchers. Resolving the challenges mentioned in this paper should make Course Builder more usable by teachers and researchers planning to offer their own MOOC.

Conclusions

MOOCs are impressive and emerging online distance education platforms that have been gaining increased popularity by the prominent educators, researchers, and universities worldwide. Google Course Builder and App Engine provide an open, free, and 24/7 accessibility for the participants of a MOOC. There are many advantages of using Course Builder for designing and delivering a MOOC but some minor challenges may emerge for first-time adopters. This paper noted a few of the issues we observed and demonstrated alternative solutions. The findings of this study will help prospective educators and researchers in designing and implementing such a MOOC. As more educators design more MOOCs using Course Builder, more interactive and additional features are likely to emerge within this platform.

Acknowledgement

The work presented in this paper was supported in part by a Google CS4HS grant and an NSF CE21 award (CNS-1240944).

References

- Cheverie, Joan. *MOOCs an Intellectual Property: Ownership and Use Rights*. Retrieved 18 April 2013, from <http://www.educause.edu/blogs/cheverij/moocs-and-intellectual-property-ownership-and-use-rights>
- David F Carr (20 August 2013). Udacity hedges on open licensing for MOOCs. *Information Week*. Retrieved 21 August 2013.
- Doukas, C., & Maglogiannis, I. (2011, November). Managing wearable sensor data through cloud computing. In *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on* (pp. 440-445). IEEE.
- Hossain, M. M. & Quinn, R. J. (2012, March). Prospective Use of Web 2.0 Technologies in Promoting Mathematics Education in the United States. In *Proceedings of Society for Information Technology & Teacher Education International Conference, 2012* (1). 3637-3642. Chesapeake, VA: AACE
- Lewin, Tamar (20 February 2013). Universities Abroad Join Partnerships on the Web. *New York Times*. Retrieved March 16, 2014.
- MOOCs Directory. (2014). *Higher Education MOOCs*. Retrieved May 5, 2014 from, http://www.moocs.co/Higher_Education_MOOCs.html
- Norvig, P. (2012). *Helping the World to Teach*. Retrieved March 14, 2014, from <http://googleresearch.blogspot.com/2012/09/helping-world-to-teach.html>
- O'Reilly, T. (2009). The war for the Web. *IT Conversations*. Retrieved October 30, 2010, from <http://itc.conversationsnetwork.org/shows/detail4317.html>
- Pappano, Laura. The Year of the MOOC. *The New York Times*. Retrieved 18 April 2014, from <http://www.nytimes.com/2012/11/04/education/edlife/massive-open-online-courses-are-multiplying-at-a-rapid-pace.html>
- Sanderson, D. (2009). *Programming Google App Engine: Build and Run Scalable Web Apps on Google's Infrastructure*. O'Reilly Media, Inc.
- Thomas, D. A., & Li, Q. (2008). From Web 2.0 to Teacher 2.0. *Computers in the Schools*, 25(3-4), 199-210.
- Wang, Y., Wang, S., & Zhou, D. (2009). Retrieving and indexing spatial data in the cloud computing environment. In *Cloud Computing* (pp. 322-331). Springer Berlin Heidelberg.
- Wilkowski, Julia. (2013). A Comparison of Five Google Online Courses. Retrieved on May 1, 2014 from <http://googleresearch.blogspot.com/2013/09/a-comparison-of-five-google-online.html>