openHPI: Evolution of a MOOC Platform from LMS to SOA

Michael Totschnig, Christian Willems and Christoph Meinel

Hasso Plattner Institute, University of Potsdam, Potsdam, Germany

Keywords: Massive Open Online Course, Learning Management System, Service Oriented Architecture.

Abstract: This article presents a new platform for Massive Open Online Courses (MOOC), developed at the Hasso Plattner Institute in Germany. After describing the evolution of the MOOC concept and the format, we explain how we defined the requirements for the platform, how we evaluated different open source learning management systems as candidate solutions, and how the actual platform was built. The results of two first courses delivered through the platform are presented and an outlook is given towards a planned redesign of the platform based on a Service Oriented Software (SOA) approach.

1 MOOC FORMAT AND HISTORY

The concept of Massive Open Online Courses (MOOC) has been coined in 2008 in the context of a course experiment conducted by Canadian educational researchers Stephen Downes and George Siemens, who opened a for credit course titled Conncectivism and Connective Knowledge at the University of Manitoba, Canada to open registration. The gist of the experiment consisted in encouraging learners to take the course content not as the end, but as the beginning of an autonomous and active journey defined by the connections the learner creates between resources and with co-learners (McAuley et al., 2010).

The same concept has recently been invested with a considerably different meaning when applied to University courses that are migrated from the setting of a closed physical classroom with a limited audience to an open online environment with a massive audience. MOOCs in this second sense have originated at Stanford and MIT and some of them attracted more than 100,000 participants.

Most discussions about MOOCs distinguish between these two concepts and formats, often referring to them as cMOOC and xMOOC (Siemens, 2012). At the heart of this distinction lies the tension between two opposing forces that are acting on higher education: On the one hand, higher education increasingly leaves the confines of national boundaries, and relates to supranational frameworks (e.g. the Europoean Bologna process) and global markets. On the other hand, pedagogical theories that conceive the learning process as active and social instead of passive and individual no longer are the privilege of alternative and marginal institutions or educators, but are now the backbone of commonly accepted learning designs like collaborative or problem-based learning. MOOCs, both the c and x variants, can be clearly situated in both movements:

- By offering a MOOC today, universities try to position themselves as global leaders of innovation, and as educational institutions capable of delivering high-quality education on a global scale. And in the same time invest in their attractiveness for the most talented prospective candidates.

- While a MOOC can be organized according to an established model of instruction, where teachers transmit a well-defined body of knowledge to unknowing learners that prove their progress by repeating back that same knowledge, it also offers the opportunity of integrating collaboration and exploration into the learning design. And even if xMOOCs have been criticized for a lack of pedagogical innovation, it can be hypothesized that their massive success is due both to the effervescent nature of their discussion forums and to the learning tools available in many MOOC platforms that allow creative exploration of the domain through virtual laboratories.

2 openHPI's COURSE FORMAT AND REQUIREMENTS

openHPI is a platform for xMOOCs, hosted at the Hasso Plattner Institute in Potsdam (HPI), Germany. openHPI is the result of tele-TASK, a research and development project conducted since 2004, which has brought into existence an advanced lecture recording system (Schillings and Meinel, 2002), and an online portal[1] for the distribution of lecture videos. While the tele-TASK portal has been augmented with sophisticated semantic web search capabilities (Sack et al., 2009) and social web functionalities, it mainly stayed focused on delivering lecture content to HPI's students allowing them to replay or to replace the class lecture. In the advent of the MOOC format, we see the opportunity to open up the pedagogical quality and domain expertise formerly reserved to our students to a broader audience.

While the inspiration for our project stemmed from the success of the Stanford courses about artificial intelligence[2] and databases[3] and the MIT course on Circuits & Electronics[4], we set our ambition to define a course format following a unique educational scenario: The subject domain is split up into six weekly units. For each week, video lectures, reading materials, and quizzes are produced and presented in a learning sequence. Discussion forums are set up for each week, and actively moderated by the teaching team. Learning progress is assessed through self-tests that can be taken an indefinite number of times, and homework, where points are granted and collected for the final score, required for obtaining the certificate.

For the technical implementation of the platform, it was clear that we did not want to rely on an external SAAS hosting solution, but create a platform we could freely adapt and evolve. Teaching at HPI focuses on IT-Systems Engineering, and the ambitious project of creating a platform for thousands of learners constitutes a very interesting challenge for our own teaching and research. Critical success factors for the fulfilment of this ambition were identified with respect to the delivery of content, the learning process, and community building.

2.1 Learning Content

The teaching team should be empowered to concentrate on the quality of the content by being provided

intuitive and powerful tools for content editing and structuring.

The presentation of the learning content should suggest a meaningful path to novice learners while giving advanced learners the freedom to jump to topics most relevant to them.

xMOOCs draw on the distinctive engagement qualities of video lectures, chunked into small-sized segments. The platform must allow to embed video content, and to enrich it with textual explanation. openHPI uses videos from the tele-TASK portal, where lecture video recordings already exist in form of chunk podcasts and additional metadata extracted from the videos.

Learning content needs to be presented in its hypertextual structure, in order to allow learners to grasp more than a linear sequence of content, i.e. the rich connections that exist between knowledge inside and across learning domains.

2.2 Learning Process

The learning environment must support the learning process by allowing learners to test new competences and by confronting them with graphical representations of their progress. Assessment tools need to be user-friendly and interactive in a way to engage and motivate learners. The synthetic representation of learning progress must be easily accessible from any part of the platform.

Learners should be able to annotate content with personal notes only available to them and with shared notes and comments that trigger reactions from the teaching team and discussions in the learning community.

Learners should also be allowed to connect the learning experience with their own tools and devices, e.g offline consumption of lecture videos, or integration of course schedules with personal productivity environments.

2.3 Learning Community

The distinguishing feature of the MOOC format is its social event character: In the past, universities have made course materials available on institutional websites, and eventually provided feedback forms or discussion possibilities, for example MIT's openCourseWare project (Lerman et al., 2008). MOOCs take place during a given time period, and hence concentrate the otherwise dispersed participation into a coherent site of collective learning. Discussion forums should allow the teaching team to trigger participations and learners to question the content.

---

[1]http://www.tele-task.de/

[2]https://www.ai-class.com/

[3]http://www.db-class.org/

[4]https://6002x.mitx.mit.edu/

Many learners are less comfortable when taking part anonymously in large groups and prefer the intimacy of small groups, eventually defined by similar characteristics like age, location, or interests. The platform should allow learners to find like-minded learners, and to define protected spaces for groups to organize a collaborative learning experience.

The platform must not lock the user into its own confines, but allow users to connect their learning experience to their social networks.

2.4 Quality Attributes

These functional requirements are mainly compatible with what many learning management systems, proprietary or open source, provide. But while weighting quality attributes (Bass et al., 2005), like stability, scalability, usability, look & feel, extensibility, maintainability, security and performance, important questions arose:

On the one hand with respect to performance: Learning management systems have been conceived for the context of schools or universities where class sizes range from tens to at the most hundreds of students. Delivering a MOOC to thousands of possibly concurrent users needs a robust technical infrastructure, and a scalable architecture.

On the other hand, we envisioned a platform that would be easily extensible in order to implement teaching methods like game-based learning, peer teaching and evaluation, and to connect to virtual laboratories.

Last but not least, we set as our goal, to attract learners through an attractive online experience that does not fall behind their experience with modern Web 2.0 platforms.

3 openHPI's TECHNICAL INFRASTRUCTURE

With the start of the openHPI project, the first task was to build up an infrastructure that scales for a potentially massive amount of users and is highly available, and to implement a suitable courseware for the desired course format.

3.1 Courseware Implementation

The first general decision to be taken was between designing and implementing an own solution based on a common web framework and the adaption and customization of an existing (open source) software project. When collecting the requirements for a

courseware platform it became obvious that the development of a courseware platform from scratch would take at least 6 months of work for a full time developer team which is a huge (financial) effort for a first course with experimental character. For this reason, we decided to build openHPI on top of an existing tool, that would allow to us to gain experience with the construction, administration and delivery of massive open online courses, and that both technically and legally allowed and facilitated experimentation with and modification of the system.

While evaluating the landscape of existing platforms for the delivery of online courses it became apparent that several types of systems had each unique advantages to offer:

- Content management systems with respect to the flexibility of the management and delivery of learning content;
- Collaborative platforms with respect to the communication features;
- Learning management systems with respect to the support for quizzes and course design.

Finally the decision was made in favor of an open source learning management system, where a number of potential candidates were evaluated. Among these candidates, the projects Sakai CLE, Sakai OAE, Canvas LMS, and Lernanta (actually a LMS framework) were investigated and matched to the openHPI requirements. The evaluation results are summarized in Table 1.

Canvas was chosen because of its modern user interface and the availability of crucial functional components necessary for implementing openHPI's course design, mainly the sophisticated quiz engine for managing practice exercises, assignments and exams, the discussion forum and the user-friendly interface for creating and presenting the course module structure.

Nevertheless, it was desired that the platform did not look or behave like an actual LMS, since LMS offer users much more freedom than openHPI needs for the provision of its courses. A developer team of 4 full time working students put 5 weeks of work into the adaption and customization of the Canvas LMS, which resulted in a massive change of the Web user interface as seen by student users. The main changes were:

- complete rebrush of UI and platform navigation;
- additional content type for learning units: flexible video player;
- content navigation adopted to openHPI's 6 week course schedule;

Table 1: Comparison of Open Source Learning Management Systems.

	Canvas LMS	Sakai CLE	Sakai OAE	Lernanta
Course Landing Page	Customizable, Rich Text Editor	Static landing page from course information	Content Repository allows flexible content	Tasks and Activity Wall, not customizable
Content Sequence	Tool for course design (modules and content blocks)	Simple syllabus tool	Hierarchic organization of content pages	Tasks can be reordered
Learning Unit Presentation	Generic blocks inside a module	As wiki pages	See landing page	A taks is defined by only one content block
Quiz Environment	Very flexible, sophisticated UI	Complex functionality, but sluggish and overloaded UI	Only available through integration with Sakaii CLE	-
Discussion Forum	Threaded discussions, no voting	No voting	No voting	Participants can post messages to activity wall and comments to tasks
Learning Progress	Users can track visited units	Tracking of submitted tests	-	-
Announcements	Rich Text Editor	Via external RSS feed	-	-
Maturity	Closed-source version used productively by schools and universities	Used by major US universities	Experimental	In active development, used productively P2PU[5]
Scalability	Ruby on Rails, needs resources	Needs resources, but works for more than 100k users	No experience	No experience
Useability, Look & Feel	Modern UI	Feels sluggish, old-fashioned	Modern UI	Clean and stylable UI
Extendability	API, vendor plug-ins possible	API	Relies on extensible framework	-

- additional navigation bar for browsing the course sequence;

- integration of a helpdesk;

- enhancement of the discussion forum (context-specific discussions, tagging, search functionality).

The adoption process kept on going during the first two courses, where the development team gathered user feedback and wishes from the community to leverage the running system.

3.2 Hardware Platform

The difficulty in building the infrastructure mainly was about estimating the resources needed and depended on a) the application-dependent demand for resources and b) the number of expected participants. Due to a strict schedule for the first course on openHPI a temporary website for registration was set up rapidly and went online about 4 weeks prior to the launch of the actual courseware platform and 5 weeks before the start of the first course.

The demand for openHPI's offering turned out to be quite overwhelming: one week after the launch of the temporary registration website, the database counted more than 10.000 unique visits and about 3.000 registrations. At the point of the turnover to the actual course platform, there were more than 10.000 registered users, of which 9.000 students enrolled for the first course, 30.000 unique visits and 110.000 page impressions.

Although these numbers were a baseline for the final user count and could help when estimating the resources, openHPI was put on a flexible and scalable infrastructure using a private cloud framework (namely OpenNebula) for the operation of the adopted Canvas LMS (see Figure 1).

The private cloud infrastructure allows flexible scalability and the provision of more computing re-
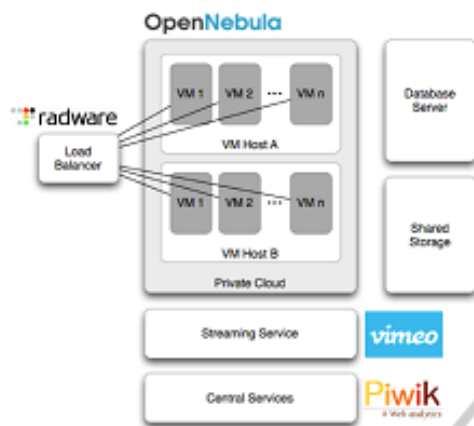
Figure 1: Overview of openHPI hardware architecture.

sources by simply adding additional physical hosts to the cloud respectively shutting down servers for reduced power consumption when the server load is low[6].

In the initial resource pool, two VM hosts were provided, each with 64 cores and 64 GB RAM as well as fast RAID 5 storage systems. Additional physical servers provide the shared database and central services like website monitoring. Incoming requests are distributed by a loadbalancer appliance that also handles the SSL sessions.

The video streaming was outsourced to the stream hosting service Vimeo, which also scales with rising demand and offers a reasonable cost model.

4 INSIGHTS FROM THE FIRST COURSES

openHPI's first two courses have met with substantial interest from the respective target audience: 13,126 learners registered for the first course, from which 4,068 actively participated and 2,137 received the graded certificate of successful completion. The second course had 9891 registered learners, with 2726 active participants, and 1635 successful completions with graded certificate. The students took about 100.000 (resp. 80.000 for the second course) self-test attempts and submitted about 18.000 (14.000) solutions of homework assignments. During the 6 weeks of course runtime, the community generated about 1.600 forum postings in the first course and more than 3.000 postings in the second course.

For the second course, we observed, that Sunday and Monday were the peak days concerning page

---

[6]For a general evaluation of cloud based services in education, see (Ivanov, 2011).

access: up to 6.000 unique visitors generated up to 60.000 page impressions. This peak behaviour was actually expectable, since Monday was the day when the weekly homework submissions were due and the course content for the next week was published. Concerning the dimension of the available hardware platform, it never came to shortage of resources that would have led to increased response time for users even though the implemented software platform turned out to be greedy for computing resources.

The basic insight after first experiences with the implemented platform for openHPI is that traditional learning management systems are actually not suitable for the operation of massive open online courses. This insight was gained from two observations. The first one is a different demand concerning scalability: traditional universities usually offer a large number of courses with a relatively low number of participants in each course (hundreds of students). In opposite to this, openHPI offers only very few courses at a time, with thousands of students in a course. This led to massive performance problems with the Canvas LMS since many standard system operations were implemented without focus on performance but on maintainability and simply did not scale with participant numbers larger than about 1.000 students. Additionally, several UI elements (i.e. where lists of users were involved) became unusable.

The second aspect is, that LMS focus on management of courses, students, tests, learning material, etc. while the focus of a platform for massive open online courses should be on social activity respectively the activity of the students in general. Learning Management Systems are designed for courses run and tutored by teachers or teaching teams, while the openHPI courses should empower the students to peer teach each other. The restricted suitability of the LMS-based implementation of the openHPI platform manifested e.g. in the non-availability of certain features for students, that have been available for teachers. I.e. a teacher could link every content item belonging to the course within related discussion postings, while students would have to manually copy & paste the respective links or describe the item in question in their own words. In general, LMS lack the support for mixed roles between course providers and course participants.

5 openHPI's NEXT GENERATION

Based on the evaluation of above shortcomings, currently a new generation for the openHPI platform is

planned. It will be based on a service-oriented architecture with the following design principles in mind:

- The new system should integrate with state-of-the-art backend services, if they are available as open source solutions and can be trusted to be both scalable and reliable. For example we evaluate the incorporation of digital asset management systems like Fedora.

- Service-oriented system design similar to component-based approaches strives at modularity and re-usability. Well-defined interfaces allow to separate the construction of the system into development tasks that can be executed by independent teams in parallel.

- Making services consumable from different client environments allows straightforward implementation of alternative client software, i.e. for mobile devices (mobile websites and native apps for iOS, Android and Windows 8).

- By decoupling the consumption of services in specific contexts from the provision of these services, both can evolve independently with respect to the affordances of their respective environment, for example the integration of new UI libraries, or the adoption of new communication protocols.

- A service-oriented architecture provides a sound basis for the integration of the learning platform into existing and future content production workflows. For example, we plan to integrate openHPI with the tele-TASK portal, were tools for collaborative work with e-lectures, such as live scripting and temporal (social) annotations have already been implemented.

- Allow exposure of services to third-party tools that provide value-added services to learners and teachers (e.g. personal learning environments, social networks or competence management (e-portfolio) systems.

- In the global competitive MOOC landscape, platforms need to provide unique learning tools in order to distinguish themselves from the competition. Hence it is important that the system can connect with external learning facilities, such as virtualized computing labs or simulation environments for hands-on training.

REFERENCES

Bass, L., Clements, P., and Kazman, R. (2005). *Software architecture in practice*. Addison-Wesley, Boston ; Munich [u.a.].

Ivanov, I. (2011). Cloud computing in education - the intersection of challenges and opportunities. In Verbraeck, A., Helfert, M., Cordeiro, J., and Shishkov, B., editors, *CSEDU 2011 - Proceedings of the 3rd International Conference on Computer Supported Education, Volume 1, Noordwijkerhout, Netherlands, 6-8 May, 2011*, page 05. SciTePress.

Lerman, S., Miyagawa, S., and Margulies, A. (2008). Opencourseware: Building a culture of sharing. In Iiyoshi, T. and Kumar, M. S. V., editors, *Opening Up Education: The Collective Advancement of Education Through Open Technology, Open Content, and Open Knowledge*, pages 213–228. MIT Press MA,, USA.

McAuley, A., Stewart, B., Siemens, G., and Cormier, D. (2010). The mooc model for digital practice. http://www.elearnspace.org/Articles/MOOC_Final.pdf. [Online; accessed 04-March-2013].

Sack, H., Baumann, B., Groß, A., and Meinel, C. (2009). Linking tele-task video portal to the semantic web. In *9th International Conference on Innovative Internet Community Systems (I2CS09)*, volume P-148 of *Lecture Notes in Informatics (LNI)*, pages 15–17, Jena, Germany. Gesellschaft für Informatik.

Schillings, V. and Meinel, C. (2002). tele-task – teleteaching anywhere solution kit. In *ACM SIGUCCS*, Providence, USA. ACM Press.

Siemens, G. (2012). Moocs are really a platform. http://www.elearnspace.org/blog/2012/07/25/moocs-are-really-a-platform/. [Online; accessed 04-March-2013].