Integration of External Tools to Foster Learner Interaction in MOOCs: The Example of Codeboard

Carlos Alario-Hoyos, Iria Estévez-Ayres, Carlos Delgado Kloos,
Raquel M. Crespo-García, Julio Villena-Román, Jorge Ruiz-Magaña

Department of Telematic Engineering, Universidad Carlos III de Madrid
Av. Universidad, 30, 28911, Leganés (Madrid), Spain
{calario, ayres, cdk, rcrespo, jvillena, jrmagana}@it.uc3m.es

Abstract. Interaction is essential to promote learner engagement in MOOCs. However, the main MOOC platforms have important limitations regarding the interactivity of the built-in tools they provide. Consequently, teachers struggle to design MOOCs that foster learner interaction with the types of educational resources available. This paper presents a successful example of integration of an existing and interactive external tool in three MOOCs on computer science deployed in the edX platform. This tool is Codeboard, a web-based development environment which allows editing, compiling and executing code in different programming languages directly from the browser. This tool is integrated in edX through the well-known interoperability standard IMS LTI, which is supported by both the edX platform and Codeboard. After running the three MOOCs with 112 Codeboard activities included, results show an extensive use of this type of activity and a very positive evaluation on the usefulness of Codeboard by the learners.

Keywords: MOOCs, programming, interaction, integration, Codeboard, edX.

1 Introduction

MOOCs (Massive Open Online Courses) must be designed in such a way that they are engaging to learners [1]. This is important for two main reasons. First, there are already many MOOCs in similar topics (more than 6800 MOOCs from more than 700 Universities until the end of 2016 [2]), and only those MOOCs which are truly appealing and have better ratings from past learners will eventually get massive numbers of enrolments [3]. Second, MOOCs demand a significant workload on the learner, who must have self-regulated learning skills to work in a fully online, barely-guided environment [4]; engaging MOOCs help to strengthen learner's extrinsic motivation [5] to be able to make the most of these courses.

One of the ways that can help to increase learner engagement in MOOCs is the promotion of interaction, both with the course materials [6], and with the peers through social tools [7]. Interaction, both in face-to-face and in online settings, has been put at the very core of learner-centered constructivist and connectivist environments for more than a decade [8].

However, most MOOCs replicate the traditional, and harshly criticized teaching model of information delivery [9], replacing lectures with short videos, and including basic activities, mainly in the form of multiple choice tests, in the evaluation system [10-11]. MOOC platforms are partially responsible for this replication of the traditional class model, as they are limited in the types of built-in tools they provide, hindering the design of more interactive MOOCs.

Nevertheless, when designing a MOOC, it is possible to consider the integration of third-party external tools as part of the course, to generate a more interactive environment. To this end, custom integration developments can be implemented, although it is also possible to make use of standards, if supported by both the tool to integrate and the MOOC platform [12]. Particularly, IMS Learning Tools Interoperability (LTI) is a widely-adopted integration standard by the IMS Global Learning Consortium that allows a seamless integration of external tools in educational platforms [13].

This paper presents the integration through the IMS LTI standard of an existing external tool, Codeboard (a web-based development environment) [14], in three MOOCs on computer science deployed in edX, with the aim to foster learner interaction. Using this tool teachers can design the skeleton of programs of incremental difficulty in several programming languages, and learners can modify, compile and run them directly from their browser, seamlessly integrated in the activity flow of learning sequences of MOOCs deployed in edX. The effect of the integration of this tool to foster learner interaction is assessed through learners' self-reported evaluations of the usefulness and difficulty of Codeboard activities, and through logs that represent the actual use of this tool.

The remainder of this paper continues with a brief overview of the three MOOCs in which Codeboard was integrated. The following section explains how activities designed in Codeboard were integrated in these three MOOCs. Next, results taken from learners' opinions and logs are presented and discussed. The paper finishes with conclusions and some future lines of work.

2 XSeries with three MOOCs on Introduction to Programming with Java

Teachers from the Departments of Telematics Engineering and Computer Science at Universidad Carlos III de Madrid (UC3M) started in 2015 the design and development of a series of three five-week MOOCs dedicated to introducing programming using Java as the driving language. These three MOOCs were released under the name "Introduction to Programming with Java" [15] and were deployed in edX as an XSeries, which is the brand that receives in this platform a series of related courses.

1. "Starting to Code with Java" was the first of the three MOOCs (MOOC 1). It ran three times (from April 2015 to June 2015 in a synchronous instructor-led mode, and from November 2015 to June 2016 and October 2016 to June 2017 in a self-paced mode). This first part introduced the basis of programming from scratch in a bottom-up approach (going from imperative programming to object-oriented programming).

2. "Writing Good Code" was the second of the three MOOCs (MOOC 2). It ran twice (from April 2016 to June 2016 in a synchronous instructor-led mode, and from November 2016 to June 2017 in a self-paced mode). It dealt with error detection and correction, also following a bottom-up approach (from debugging and testing of code to high-level design of programs).
3. "Fundamental Data Structures and Algorithms" was the last of the three MOOCs (MOOC 3). It ran once (from April 2017 to June 2017 in a synchronous instructor-led mode). This third part addressed linear and non-linear data structures and basic algorithms which can be applied on them, such as insertion, searching or sorting of data.

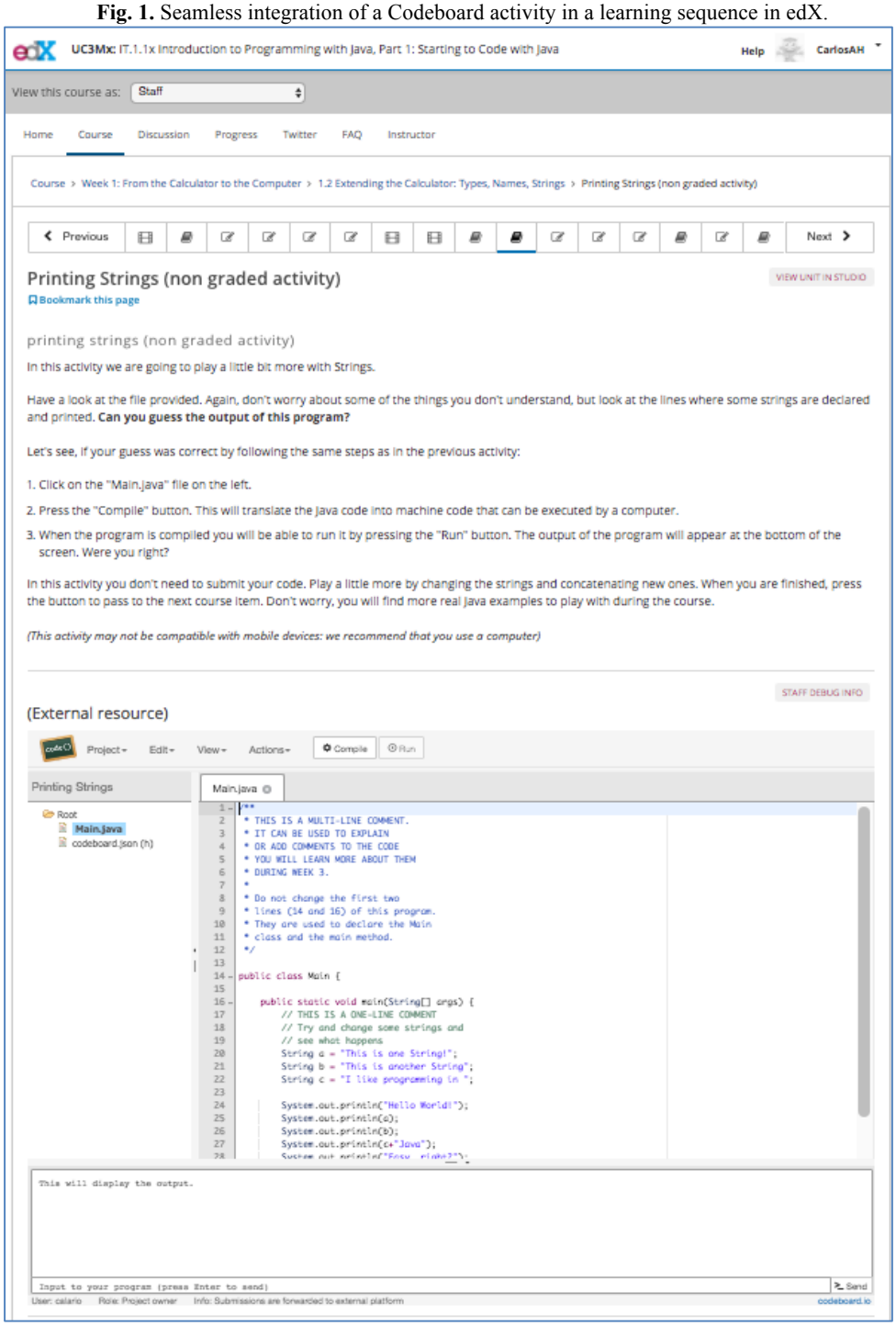
These three MOOCs were offered in English (videos also contain subtitles in Spanish), and got more than 300,000 enrollees overall. Learners expected workload is between 5 and 10 hours per week. Videos were an important part of the learning materials, although these MOOCs were designed to include numerous interactive activities, both using edX built-in tools, and externals tools, such as Codeboard. Each week contained four learning sequences, whose core were videos and interactive activities, a laboratory, which intended to propose a transversal development project for each MOOC, and summative activities, which were used to calculate learners' grade in the course. A detailed description of the design of the course can be seen in [16].

3 Integration of Codeboard activities in the MOOCs

The integration of Codeboard, more specifically the integration of activities in which learners must modify, compile and run a given Java code using Codeboard, has been possible because both Codeboard and edX support the IMS LTI standard. No specific developments were needed to integrate Codeboard activities in the MOOCs on "Introduction to Programming with Java". However, some configuration steps were needed in edX before learners could use Codeboard. First, LTI had to be enabled in the advanced configuration settings of the MOOC in edX, adding two values for parameters key and secret (provided by Codeboard staff), to identify requests between the platform and the tool. Then, LTI activities were created where appropriate in the course, given them a name and a URL; each Codeboard activity created has a unique URL, which had to be copied and pasted in the corresponding LTI activity in edX.

The integration of Codeboard in edX is seamless for learners (see Figure 1), as Codeboard activities are shown within the learning sequence. Each learning sequence contains videos, exercises and others (under which category Codeboard activities fall), as it can be seen in the three types of icons of the top bar in Figure 1. Each Codeboard activity has an introductory text explaining the activity; this text is created with a built-in HTML component in edX; Codeboard activity is located below the text. Learners can modify the code, compile and run it as many times as they wish. However, every time learners refresh the page they go to the initial state of the activity (the default code provided by the teachers). Learners can, however, create an account in Codeboard, and if they are logged in, they will be able to save their progress. Codeboard activities aim to practice and are not mandatory to pass the course.

Fig. 1. Seamless integration of a Codeboard activity in a learning sequence in edX.



Codeboard activities are distributed throughout the five weeks of each MOOC. Particularly, in each week, they can be found in any of the four learning sequences, or in the laboratory, which is a special additional learning sequence in which learners work in a project which is transversal to the MOOC, has incremental difficulty, and focuses on the specific concepts addressed in that week. Table 1 shows the distribution of Codeboard activities in the three MOOCs, indicating the number of Codeboard activities in the four learning sequences that make up each week and in the laboratory.

Several aspects are worth noting looking at Table 1. First, there are many Codeboard activities throughout the three MOOCs. This was an intentional design decision to foster learner interaction. Only week 5 of MOOC 2 does not include any Codeboard activity. The reason is that this week is dedicated to introducing ethical issues in software development, thus including other types of activities. Second, apart from Codeboard, other external tools were used in these MOOCs, some of them seamlessly integrated in edX, such as Blockly [16], and some of them downloaded and installed in learners' equipment, such as Eclipse or Greenfoot [16]. This explains why some weeks in the MOOCs have a lower number of Codeboard activities, precisely because other external tools were used more intensively instead. For example, Blockly was the primary tool used in the laboratory of week 1 of MOOC 1 to practice basic programming concepts through visual games. In addition, learning sequences in week 2 of MOOC 2 mainly relied on Eclipse to explain testing of programs using the built-in JUnit (a popular framework for testing) features Eclipse has. Third, Codeboard was the main tool for most laboratories. In the first MOOC teachers facilitated the realization of the laboratory splitting the work in several Codeboard activities, leading to a complete solution at the end of each week's lab. In MOOCs 2 and 3 the laboratory was designed to have one or two large exercises in a single Codeboard activity.

Table 1. Codeboard activities integrated in the MOOCs.

MOOC	Week	Number of Codeboard activities in the four main learning sequences	Number of Codeboard activities in the laboratory
Part 1	1	8	0
	2	2	4
	3	7	3
	4	1	8
	5	1	4
	Total	***19***	***19***
Part 2	1	3	2
	2	0	1
	3	10	2
	4	2	1
	5	0	0
	Total	***15***	***6***
Part 3	1	9	1
	2	7	1
	3	7	1
	4	8	1
	5	17	1
	Total	***48***	***5***
TOTAL		***112***	

4 Results

Learners who reached the end of any of the three MOOCs had the possibility of answering an optional questionnaire in which they were asked about different aspects of that specific MOOC. Regarding Codeboard, learners were asked, in a scale from 1 to 5, about the usefulness of Codeboard activities to check their learning process (5 – very useful), and about the difficulty level of Codeboard activities (5 – very difficult). Figures 2 and 3 show the results for each of the three MOOCs. Regarding the usefulness of Codeboard activities, MOOC 1 obtained an average of 4.12 (std. 1.024, N=1075), MOOC 2 obtained an average of 4.04 (std. 1.004, N=137), and MOOC 3 obtained an average of 4.6 (std. 0.699, N=10). Regarding the difficulty of Codeboard activities, MOOC 1 obtained an average of 3.42 (std. 0.995, N=1075), MOOC 2 obtained an average of 3.29 (std. 0.925, N=137), MOOC 3 obtained an average of 4.1 (std. 0.738, N=10).

Fig. 2. Usefulness of Codeboard activities in MOOC 1 (top) (N=1075), MOOC 2 (middle) (N=137), MOOC 3 (bottom) (N=10), from 1 (low) to 5 (high) (Y-axis) and number of answers (X-axis).

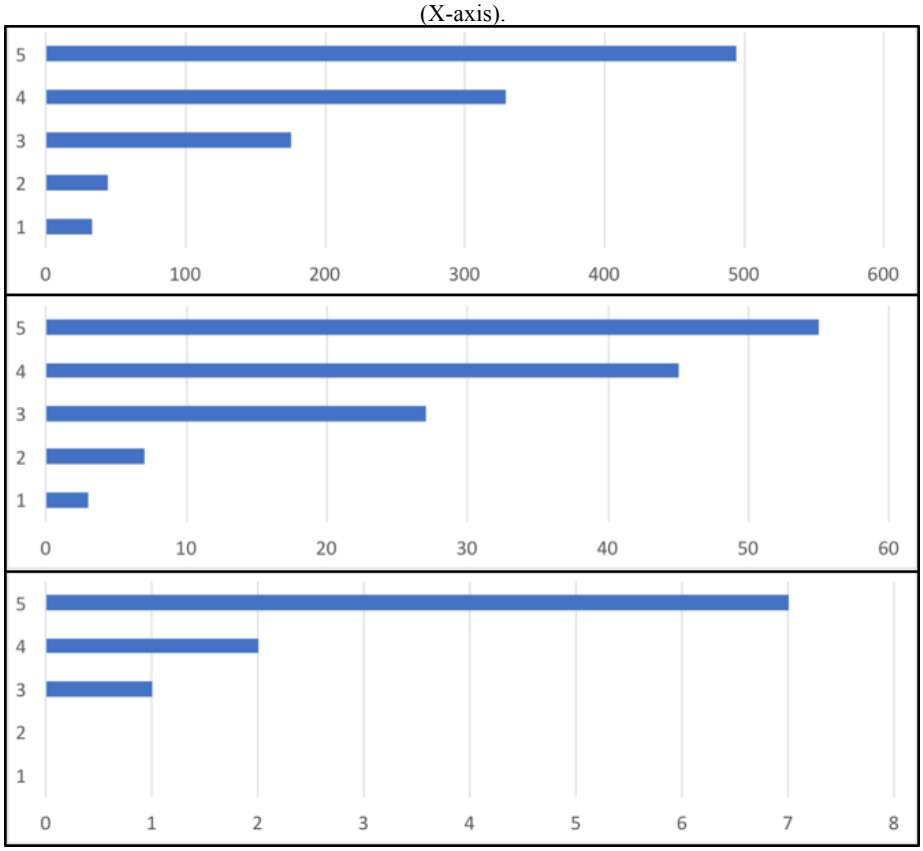
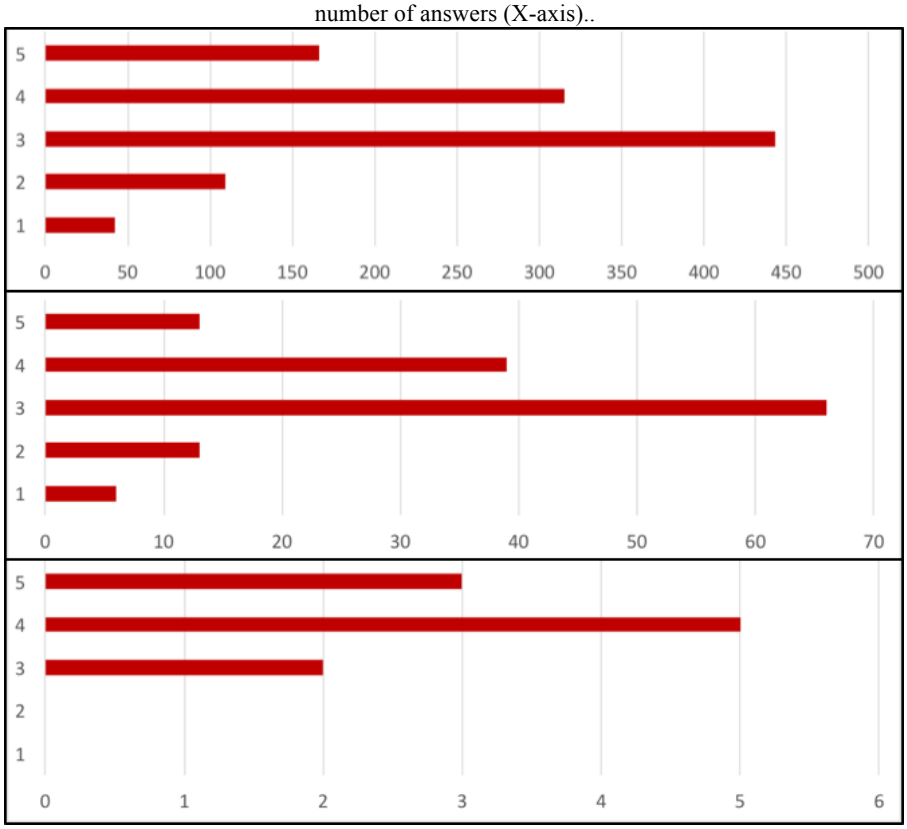


Fig. 3. Difficulty of Codeboard activities in MOOC 1 (top) (N=1075), MOOC 2 (middle) (N=137), MOOC 3 (bottom) (N=10), from 1 (very easy) to 5 (very difficult) (Y-axis) and number of answers (X-axis)..



In these results, it stands out the decreasing sample size in the three MOOCs (an order of magnitude less in each subsequent MOOC). The explanation to this relies on several facts. First, these are the answers accumulated in all the editions a MOOC has been running (three runs for MOOC 1, two runs for MOOC 2, and one run for MOOC 3). Second, the number of enrolments significantly decreases comparing MOOC 1 and MOOC 2 and comparing MOOC 2 and MOOC 3. Third, a low number of learners (roughly a 5% of enrollees) reaches the end of this kind of courses. And, finally, filling in the questionnaire was an optional activity.

The effect on learners of integrating this tool in the MOOCs to foster interaction is positive, since the results indicate that learners consider Codeboard activities useful. Moreover, the level of difficulty of Codeboard activities, in general, is not a factor which contributes to learner frustration (which may happen if difficulty is very high), but is moderately challenging to engage learners.

4.1 Example of use of Codeboard activities

Codeboard is also especially useful for MOOC teachers, allowing them to analyze in detail the interaction of learners with each Codeboard activity. The internal analytics provided by Codeboard shows the number of compilations and runs of code for each Codeboard activity, and their distribution over time. This allows MOOC teachers to detect if learners get stuck (i.e. Codeboard shows a high number of compilations but few executions of code), and take corrective actions, such as providing complementary explanations in the forums, or in the text that precedes the code.

Figure 4 shows an example of the information provided by Codeboard for one of the activities of MOOC 1. It represents the number of compilations (60,559) and executions (51,714) over time, with the three runs of MOOC 1 clearly framed; there is some activity between runs, as enrollees could keep working in the course, but no more new enrollments were allowed. The code provided by the teachers is simple and compiles correctly printing the resulting output on screen. No major problems are detected in this exercise due to the balanced relationship between number of compilations and runs, which is detailed in Figure 4 (bottom) for the first run of MOOC 1.

Fig. 4. Example of Codeboard activity from MOOC 1: full time span of the three runs (top) marked with red boxes, and time span of the first run (bottom).

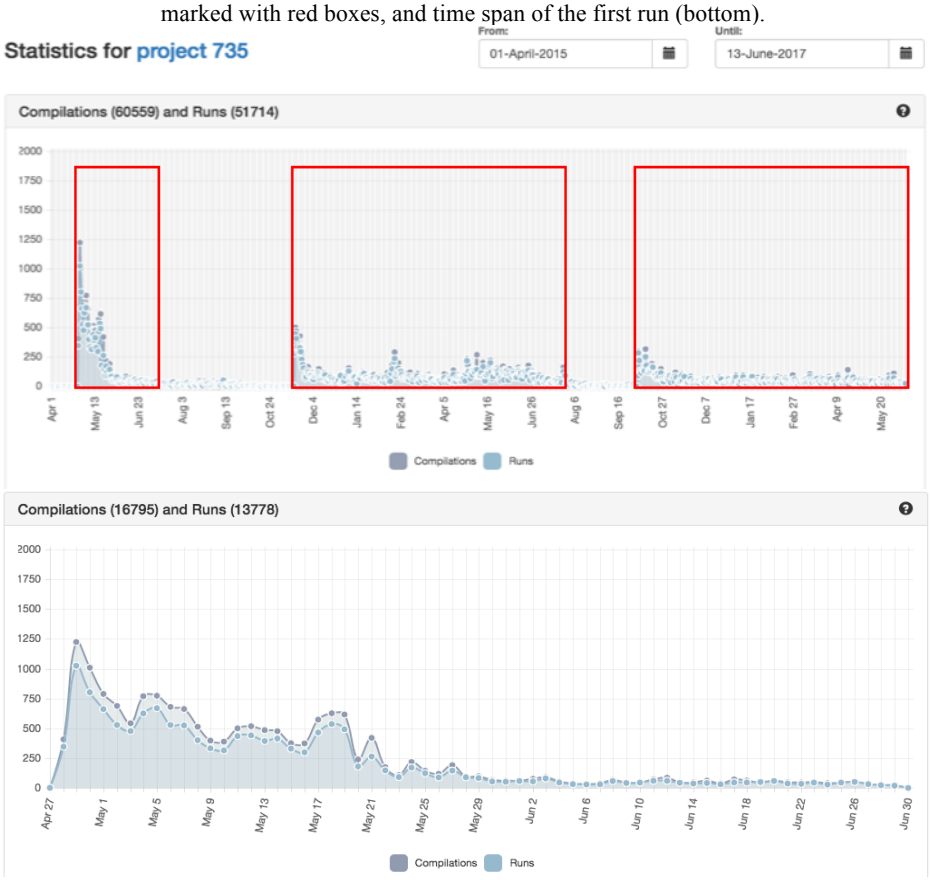
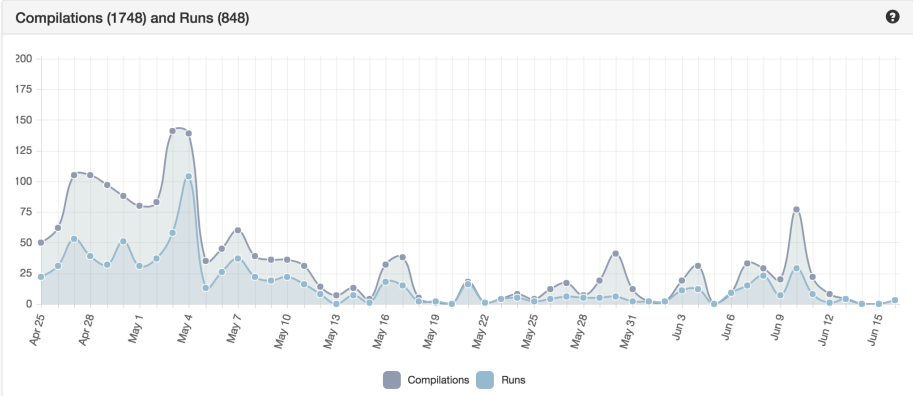


Figure 5 shows another example in which an unbalanced relationship is seen between number of compilations (1748) and executions (848) (more than twice as many compilations). This example belongs to a Codeboard activity of the first week of the MOOC 3. Teachers should therefore provide additional clues to help to resolve the exercise, as an important number of learners got stuck and were not able to advance.

Fig. 5. Example of Codeboard activity from MOOC 3: time span of the first run.



5 Conclusions

This paper presented the integration of the Codeboard tool, through the IMS LTI standard, into three MOOCs on "Introduction to programming with Java", deployed in the edX platform. The integration process was simple, since this standard is supported by both the Codeboard tool and the edX platform, and only high-level configuration adjustments were required. The aim of this integration was to foster learner interaction in an eminently practical MOOC, and where the MOOC platform did not provide purpose-specific tools for the subject taught. The fostering of learner interaction has as its ultimate goal increasing the overall engagement with the course. For this, 112 Codeboard activities were specifically created; learners positively evaluated these activities. All in all, MOOC teachers should explore how best to foster learner interaction and take advantage of existing external tools that can be integrated in the MOOC platform with this purpose.

Acknowledgements. This work has been co-funded by the Erasmus+ Programme of the European Union, projects MOOC-Maker (561533-EPP-1-2015-1-ES-EPPKA2-CBHE-JP), SHEILA (562080-EPP-1-2015-BE-EPPKA3-PI-FORWARD), COMPETEN-SEA (574212-EPP-1-2016-1- NL-EPPKA2-CBHE-JP), and COMPASS (2015-1-EL01-KA203-014033), by the Madrid Regional Government, through the eMadrid Excellence Network (S2013/ICE-2715), and by the Spanish Ministry of Economy and Competitiveness, project RESET (TIN2014-53199-C3-1-R). We especially thank Hans-Christian Estler and Martin Nordio for allowing us to use Codeboard in our MOOCs selflessly; this work could not have been done without their support.

References

1. Hew, K. F. (2016). Promoting engagement in online courses: What strategies can we learn from three highly rated MOOCS. *British Journal of Educational Technology, 47*(2), 320-341.
2. Class Central (2016). By The Numbers: MOOCs in 2016. Retrieved from: https://www.class-central.com/report/mooc-stats-2016/
3. Yousef, A. M. F., Chatti, M. A., Schroeder, U., & Wosnitza, M. (2014). What drives a successful MOOC? An empirical examination of criteria to assure design quality of MOOCs. In *Proceedings of the 2014 IEEE 14th International Conference on Advanced Learning Technologies (ICALT)*, (pp. 44-48). IEEE.
4. Zheng, S., Rosson, M. B., Shih, P. C., & Carroll, J. M. (2015). Understanding student motivation, behaviors and perceptions in MOOCs. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing* (pp. 1882-1895). ACM.
5. Alario-Hoyos, C., Estévez-Ayres, I., Pérez-Sanagustín, M., Kloos, C. D., & Fernández-Panadero, C. (2017). Understanding Learners' Motivation and Learning Strategies in MOOCs. *The International Review of Research in Open and Distributed Learning, 18*(3), 119-137.
6. Iniesto, F. (2017). User-centered design strategies for massive open online courses (MOOCs), *Open Learning, 32*(2), 188-190.
7. Zhang, D. J., Allon, G., & Van Mieghem, J. A. (2016). Does Social Interaction Improve Learning Outcomes? Field Evidence from Massive Open Online Courses (MOOCs), *Manufacturing & Service Operations Management (in press)*, 1-36.
8. Beldarrain, Y. (2006). Distance education trends: Integrating new technologies to foster student interaction and collaboration. *Distance education, 27*(2), 139-153.
9. Armellini, A., & Padilla Rodriguez, B. C. (2016). Are Massive Open Online Courses (MOOCs) pedagogically innovative?. *Journal of Interactive Online Learning, 14*(1), 17-28.
10. Kay, J., Reimann, P., Diebold, E., & Kummerfeld, B. (2013). MOOCs: So Many Learners, So Much Potential... *IEEE Intelligent Systems, 28*(3), 70-77.
11. Kaplan, A. M., & Haenlein, M. (2016). Higher education and the digital revolution: About MOOCs, SPOCs, social media, and the Cookie Monster. *Business Horizons, 59*(4), 441-450.
12. Alario-Hoyos, C., & Wilson, S. (2010). Comparison of the main alternatives to the integration of external tools in different platforms. In *Proceedings of the International Conference of Education, Research and Innovation, ICERI* (pp. 3466-3476).
13. IMS Global Learning Consortium (2014), Learning Tools Interoperability 2.0. Retrived from: http://www.imsglobal.org/lti/
14. Codeboard. Retrived from: https://codeboard.io
15. XSeries "Introduction to Programming with Java", edX. Retrieved from https://www.edx.org/xseries/introduction-programming-java
16. Alario-Hoyos, C., Delgado Kloos, C., Estévez-Ayres, I., Fernández-Panadero, C., Blasco, J., Pastrana, S., & Villena-Román, J.: Interactive activities: the key to learning programming with MOOCs. *Proceedings of the Fourth European Stakeholder Summit, EMOOCS 2016*, 319-328, Graz, Austria, 2016.