

On the way for materializing cMOOC requirements: an experience dealing with plugins on Moodle Platform

Filipe Garrido¹, Neyla Fontan¹, Felipe Abreu¹, Rita Suzana Pitangueira Maciel¹

¹Departamento de Ciência da Computação (DCC)
Instituto de Matemática e Estatística (IME)
Universidade Federal da Bahia (UFBA)
Salvador - BA - Brasil

{filipe.garrido, neyla.fontan, rita.suzana}@ufba.br
felipe.abreu@live.com

Abstract. *MOOCs with Connectivist approach (cMOOC) do not have yet a consensus about requirements for supporting environments. Therefore, this paper presents an experience for materializing cMOOC requirements through Moodle and plugins for its design. The translation of user requirements, related to the four connectivism domains, into system requirements guided the choice of Moodle plugins. This process of translation resulted in a list of software requirements and plugins for Moodle to build a cMOOC. In the future, it is expected to analyze the interactions of these learners and to verify the learning of the content proposed in the cMOOC.*

1. Introduction

MOOC is an educational environment which provides open and free registration, publicly shared curriculum, and generating results with unpredictable endings. In addition, an initial MOOC design incorporated social networking and accessible online resources such as a discussion forum facilitated by experts in the field of study of the offered course. More significantly, MOOCs are built through the engagement of learners, who self-organize their participation in accordance with their learning objectives, prior knowledge and common interests [McAuley *et al.* 2010].

However, over the years there was a departure from the original conception of the MOOC. Previously characterized as an option for access to the major centers of higher education in the world for continuous improvement of knowledge and collaborative learning, it has now become a specific type of online course to accommodate thousands of students who observe (attend) classes and carry out activities, in many individual cases, for the purpose of obtaining certification [Grainger 2013].

Nowadays, there are two basic MOOC classifications, based on different pedagogical approaches and organizational models: cMOOC and xMOOC [*ibidem*]. Based on the connectivist approach, cMOOC is grounded on the connections between the members and the sharing content and knowledge sharing among students enrolled in the course. Thus, by encouraging students to expand and create knowledge from their own interests and network searches [Mota and Inmorato 2012].

xMOOC follows a business model based on traditional online education formats. Commonly is strongly supported by multimedia materials, especially video, uses quizzes and forum to evaluate the apprentices of the courses. It follows a structured and traditional format, which follows a flow, from pre-defined knowledge by the teacher, which shares content for many students [*ibidem*].

Including xMOOCs, usually, online course environments are built from custom requirements already cataloged for this domain [Lucena *et al.* 1998; Neto, Gomes and Tedesco 2003; Grüzner and Paech 2005]. However, there is no consensus on the software requirements that a cMOOC domain system should attend. Building a cMOOC environment prototype makes it possible to identify requirements.

Prototyping is an interesting strategy for requirements elicitation, conceives an environment to validate the software engineer's interpretation of software users requirements as well as to obtain new ones. The dynamic behavior of an interface and its possible interactions to the user can be better understood through prototyping. [Bourque and Fairlwy 2014].

Therefore, the main question arises: *what are the software requirements that satisfy the connectivist practices for a MOOC course?* Additionally, *could current software platforms fulfill these requirements?* In this sense, this paper aims to present a cMOOC developed with Moodle (Modular Object-Oriented Dynamic Learning Environment) through plugins to reach user requirements from the perspective of connectivism.

The Moodle is one of the most widely used learning management system (LMS) to support distance courses and face-to-face in federal universities in Brazil [Sabbatini 2007]. Conceived as a free software platform, it has a community of developers and technological contributors who have created plugins for the most diverse needs [Dougiamas and Taylor 2003]. These plugins are a software component that adds or expands a specific feature to a complete software. It also enables the customization of interface or features depending on users' needs, such as accessibility.

We believe that Moodle, considering the version evaluated, is a platform that can serve as the basis for a MOOC. Indicate and weave guidelines for their customization to reach this goal [Do Rêgo, Garrido e Matos 2018]. However, it lacks additions to give suitable computational resources for a cMOOC. We emphasize that Moodle can vary in interface design and positioning or hide some elements of interaction, or even tools and functionalities available depending on the version. However, even if it presents variations, fundamental elements are similar between the installations of Moodle.

In the design of a MOOC system interaction is a sensitive part for the student (user) throughout their instructional path. A systematic mapping by Garrido [Garrido *et al.* 2018] presented studies that dealt with this aspect. However, as it is a functionality not foreseen among the requirements, the system interaction will not be the goal of this project.

The MOOC system engineered by a set of plugins selected to compose the prototype and validate through tests the user requirements converted into software

requirements from the connectivist perspective (cMOOC) follows the principle of Prototyping Evaluation Techniques. It aims to validate a target set of requirements and could be served as a template for future software, thereby reducing development effort [Bourque and Fairlwy 2014].

The paper is divided into five sections. In addition to the previous section, Introduction (I), followed by the section Theoretical foundation (II), methodology (III) that describes the methodological steps to implement this paper. Initiated by the identification of user requirements (A), identification of plugins (B), requirements list (C), plugins identified in the official Moodle repository (D) and cMOOC (E) environment design. Section IV and V, Results, Conclusion and Future Work, present the findings, Inferences about the developed environment, as well as its potentialities and the studies that will give continuity to this study, respectively.

2. Theoretical Foundation

As there is no consensus on requirements for cMOOC have to meet, it can be decisive for the success or failure of a cMOOC project. Defining requirements as a system characteristic or description of something the system can accomplish to achieve its goals [Sommerville 2007].

Requirements engineering is a process of discovering, analyzing, documenting and verifying the functions and constraints that the system must have [Sommerville 2007]. In the context of the development of educational systems requirements engineering has become even more challenging to deal with the teaching and learning process [Cruz Neto, Gomes and Tedesco 2003].

The system requirements are expected to contribute positively to the achievement of users' goals attention to connectivism efficiently and effectively manner. Software requirements can be divided between function and non-functional. These requirements conjecture elementary functionalities of the software to be developed in relation to the identified needs, such as meeting a particular learning theory, *e.g.* connectivism.

The connectivism is a theory of learning geared to the digital age which adapts the current technological reality and to the network society provided by the use of technological resources of Web 2.0. Knowledge is available through connection networks and learning is the ability to build knowledge from an extensive network of established connections [Siemens 2005].

The four dimensions characterized by [Downes 2008] for a connectivist learning network are essential to identify the user requirements that must be achieved by the system. Are they:

- **autonomy** - each participant manages and works toward their own sense of values and goals;
- **diversity** - participants are different from each other, and there must be different perspectives and points of view to arrive at any new understanding;
- **openness** - adherence of participants to the community is fluid and they should remain open to new members and materials;
- **interactivity** - knowledge is generated by the interaction of its members.

User requirements are usually described in a higher level of abstraction because they reflect needs described by stakeholders users who do not have the necessary technical knowledge about technology solutions. User requirements need to be detailed in software requirements that describe features to be met by the software being designed and developed. They should also consider constraints for development. Therefore, software requirements represent the link of understanding between users and developers about the system being built.

3. Methodology

The methodology of this study was divided into three phases: (i) identification of the functionalities in existing cMOOC platforms; (ii) identification of the plugins available in the Moodle platform; and (iii) designing a cMOOC environment using Moodle as a basis.

3.1. Identification of functionalities

cMOOC could be seen as an application domain because it demands specific functional and non-functional requirements. Functionalities related to the practical application of the Connectivist Theory under the perspective (requirements) of cMOOC users need to be translated into software requirements. Support the connectivist approach in the construction of solutions (systems) for this type of course should be carefully considered.

In order to identify the potentials user requirements for system development in the cMOOC domain, we analyzed: (i) papers in the literature containing definitions and characterization of MOOC; (ii) papers in the literature containing definitions and characterization of the connectivist approach; (iii) recent cMOOC, available on the web, with support for the connectivist approach. The user requirements identified in the literature and the functionalities identified in existing cMOOC was mapped into a preliminary list of software requirements.

The representation of software requirements in list served an epistemic tool for the developer team. The software requirements have been cataloged in a guide that can aid the development of cMOOC systems by reusing or customizing the proposed requirements (Figure 1).

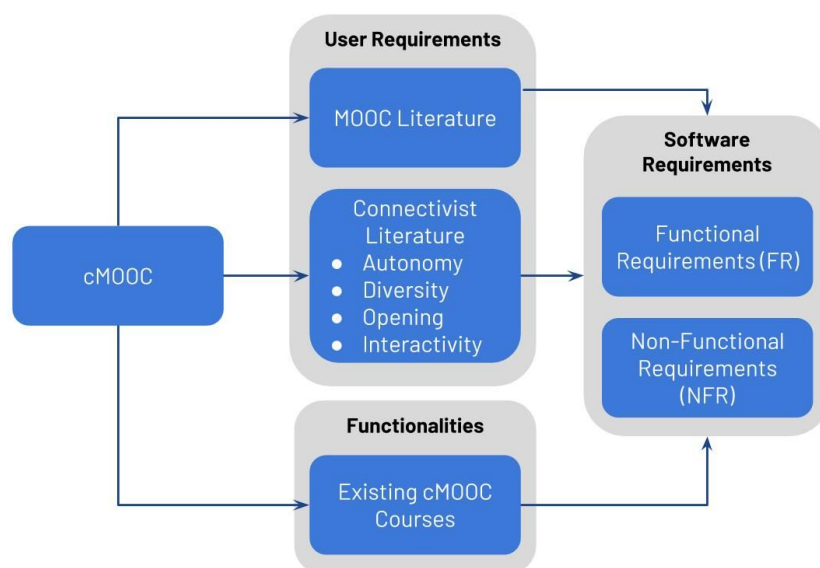


Figure 1. cMOOC - requirements, functionalities and connectivism

To identify functionalities in existing cMOOC courses, we analyzed 20 courses available on the web from 2008 to 2017. The courses analyzed includes the oldest ones that were made available in 2008 and were precursors of the connectivist approach, to more recent courses made available in 2017. Table 1 presents the list of cMOOCs analyzed, containing the name of the course, the year of its execution and the promoting institution. We highlight that in this table were chosen only the platforms (systems) indicated as cMOOC. Excluding courses in foggy environments without a proper system, such as the course promoted by Downes and Siemens to present connectivity. This course used the wiki, blogs and forum to integrate students.

Table 1. cMOOCs between 2010 and 2017

Course Name	Year	Institution
EduMOOC	2011	University of Illinois Springfield
LAK12	2012	Society for Learning Analytics Research
OPCO12	2012	Goethe University Frankfurt
MMC13	2013	University of Applied Sciences
PLENK 2010	2010	Athabasca University
ETMOOC	2013	University of Regina
FSLT12	2012	Oxford Brookes University
DS106	2017	University of Mary Washington
EC & I 831	2011	University of Regina
CMC11	2011	SUNY Empire State College
CCK12	2012	University of Manitoba
Games MOOC	2015	Center4EduPunx
CFHE 12	2012	Athabasca University
LOER 12	2012	Faculty Ecommons
OLDS MOOC	2013	JISC (Joint Information Systems Committee)
OCTEL	2014	Association for Learning Technology
Think Tank	2013	European Commission

ExIF 13	2013	FernUniversität in Hagen
MM: Dark Underbelly	2014	Duke University
MOOC Língua Portuguesa	2013	ABMES - Associação Brasileira de Mantenedoras de Ensino Superior

3.2. Identification of Plugins

The plugins were searched at the official Moodle.org¹ base to extend Moodle functionality to meet the requirements of the software as per the requirements of the user requirements. The repository has more than 1500 (one thousand and five hundred) plugins, all open-source and with a highly active developer community. We recommended that Moodle system administrators make adaptations and translations when necessary and make them available to the community.

4. Results

The results obtained are presented as follow, starting with the presentation of the requirements identified in the existing cMOOC platforms. Subsequently, the selected plugins that meet the identified requirements are presented. Finally, the design of the prototype cMOOC environment, through Moodle extended with the plugins are presented.

4.1. List of requirements

In the guide proposed by this study, user requirements have been translated into a preliminary list of software requirements for the cMOOC domain and categorized by their Functional Requirements (FR). The FR type software requirements were grouped into the four dimensions that characterize the connectivist dynamics in a learning network [14]: autonomy, diversity, openness and interactivity. These requirements are available at the link <http://bit.ly/2VJyybY>.

We emphasize that FRs specify functions that the system must be able to perform. These are software requirements that define the behavior of the cMOOC, *i.e.* the process or execution take actions on the entries of the trainees in accordance with the connectivist principles. These requirements capture (attend) the functionality from the point of view of the (requirements of) users (learners).

4.2. Plugins identified

Below are the plugins that support the requirements identified in the previous phase. The plugins will be listed with a brief explanation about each:

- **Hot Question:** Used to better understand the topics about which a group collectively wants to know more. Students can submit questions related to the topics presented in the course, which appear in an ordered list (recent questions appear first). Students then "sort" their peers' questions by clicking on a thumbs up icon - this provides the "warmth" of the question. The more votes, the hotter the question is the higher the list, it will appear.

¹ <https://moodle.org/plugins/>

- **MoodleOverflow**: A non-linear and non-chronological discussion forum. Similar to forum activity, but focuses on simple question-and-answer discussions. In addition, users rank posts and get a rating ("reputation") ranking by other students. Students who have started a discussion can mark a response as helpful; and teachers can mark a post as a (correct) solution to the question put.
- **Turnitin**: for plagiarism detection.
- **Wiki**: This is an alternative wiki to the Moodle standard. The main intention of this plugin is to provide a simple and appropriate teaching tool to students' use with the minimum of training required. Encouraging the collaborative construction of knowledge.
- **Grade reports**: It provides students with the most important part of an examination / questionnaire, that is, analysis. Each attempt each user makes on a test is examined and relevant data is portrayed through visually appealing graphs.
- **Structured feedback**: Providing structured feedback allows teachers to define any number of criteria to provide individual feedback for student task submission. They encouraging peer review.
- **Anders Pink**: Organizes, controls and displays relevant external content to the environment on any topic, along with its programs and courses. Content can be updated automatically every hour or by manual configuration.
- **Assignment Review**: Allows reviewers to publish "global" comments with attribution. Displays assignment-chained comments on assignment pages. It also allows the insertion of key markers or text that can be selected in queries to the database.

4.3. Conception of cMOOC

cMOOC system was designed as a course called Plugin Tests (Figure 2) in Moodle 3.4.6. Inaugurated on August 14, 2018, no defined deadline and the use of the selected plugins, 3 (three) dummy users were created for testing.

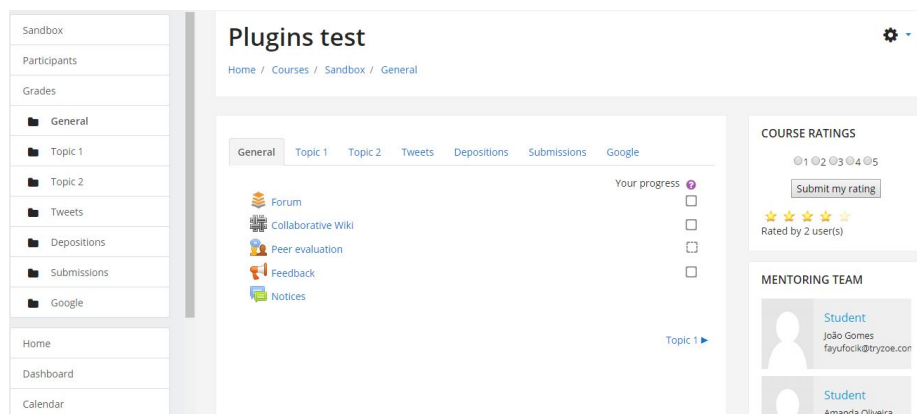


Figure 2. Home of cMOOC - Test of plugins.

The integration with Twitter allows the crossing of sources external to the course, as in the example of @BBC (Figure 3). The main effect is to use the resources of a familiar system to students (users), who can use their own profiles, and allow a constant and qualified update of content for the course, through selected content

sources.

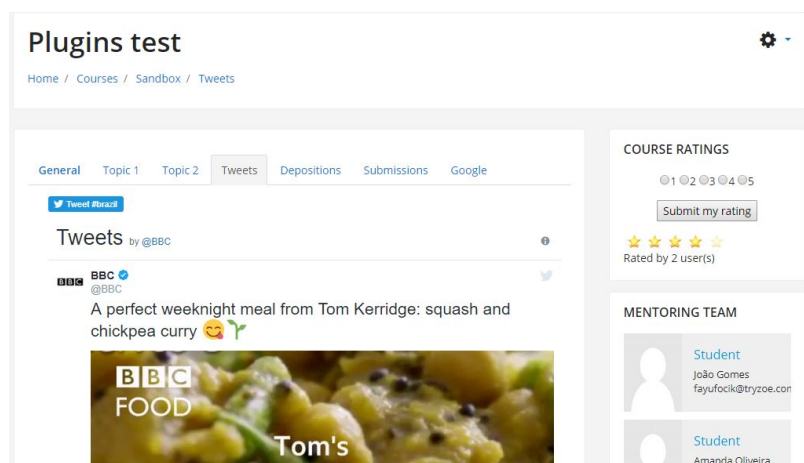


Figure 3. Anders Pink - tweets embedded in cMOOC (e.g. @BBC).

Fictitious users created test the functionality of peer evaluation, in attention to the domain of autonomy and interactivity (Figure 4). Each of the users submitted an activity and evaluated the sending of the others. With only a small number of students, only three, all evaluated all activities. In case of a greater number, tens or hundreds, the evaluation occurs by random lottery. That is, any student can evaluate another one without selection criteria.

This functionality allows the achieve all the dimensions that characterize the connectivist dynamics in a learning network: autonomy, diversity, opening, and interactivity [Downes 2008]. The same goes for the discussion forum, in our case adopting MoodleOverflow instead of the standard Moodle forum.

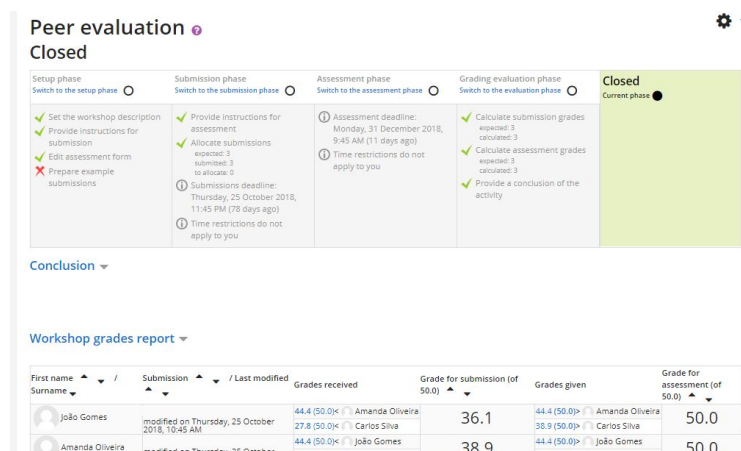


Figure 4. Peer evaluation (closed) - score and feedback among peers.

5. Conclusion and Future Works

The purpose of this study was to present a proposal of a cMOOC conceived through Moodle extended with plugins to meet users' requirements from the perspective of

Connectivism. It also enabled the creation of a list of functional and nonfunctional requirements to guide stakeholders in adapting or creating cMOOC.

Eliciting software requirements from the perspective of user requirements for connectivism service required a thorough search of the thousands of plugins available in the official Moodle repository. The set of selected plugins can be understood as the most suitable to favor communication, collaboration and knowledge construction among peers, a sine qua non for connectivity.

In addition, we hoped that the guide will foster and facilitate the theoretical-practical discussion about connectivism in MOOC environments and the academic-scientific products of the different theoretical currents of learning.

As future work, we expect to execute a cMOOC during the first half of 2019 with the content yet to be defined. Throughout this course we will analyze the interactions between students and the use of the designed environment, including social networks used externally to cMOOC. Since there is no fixed period of the beginning and end of the course, the analysis will take place in a period between 60 and 90 days.

We also intend to elicit non-functional requirements with the goal of developing an even more appropriate cMOOC user needs in a connectivist environment

References

- Bourque, P., Fairley, R. E. (2014). Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0. IEEE Computer Society Press.
- Cruz Neto, G. G. da, Gomes, A. S., Tedesco, P. (2003). Elicitação de Requisitos de Sistemas Colaborativos de Aprendizagem Centrada na Atividade de Grupo." In Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE), vol. 1, no. 1, pp. 317-326.
- Do Rêgo, B. B., Garrido, F. A., de Souza Matos, E. (2018). Moodle como ambiente MOOC: orientações para o redesign de interação. *RENTE*, 16(1).
- Dougiamas, M., Taylor, P. (2003) Moodle: Using learning communities to create an open source course management system. In: *EdMedia: World Conference on Educational Media and Technology*, 2003, Honolulu/Hawaii (USA). Anais. Waynesville: Association for the Advancement of Computing in Education (AACE) pp. 171-178.
- Downes. S. (2008) *An Introduction to Connective Knowledge*, [S.l.: s.n.]. 77–102 p. ISSN 1098-6596. ISBN 9788578110796.
- Garrido, F., do Rêgo, B. B., Maciel, R. S. P., Matos, E. (2018). Uma abordagem de design para MOOC: investigação da articulação do design instrucional & de interação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, vol. 29, no. 1, p. 388.
- Grainger, B. (2013). Introduction to moocs: avalanche, illusion or augmentation. Policy Brief - UNESCO. Institute for Information Technologies in Education. Available in: <<http://iite.unesco.org/pics/publications/en/files/3214722.pdf>>. Access in: 03 apr. 2018.

- Grüzner, I. and Paech, B. (2005). Requirements Engineering for Courseware Development. In: J.L. Mate, A. Silva, (eds.), Requirements Engineering for Sociotechnical Systems, Idea, Madrid, pp. 170-188.
- Lucena, C. J. De, Fuks, H., Milidiú, R., Laufer, C., Blois, M., Choren, R., Torres, V., Daflon, L. (1998). AulaNet: helping teachers to do their homework. In Multimedia Computer Techniques in Engineering Education Workshop. Latin American Academic Training n. ALR/B7-3011/94.04-4.0161, pp. 16-30.
- Mota, R., Inmorato, A. (2012). MOOC, uma revolução em curso. Jornal da Ciência. Vol. 26, pp. 11-12.
- McAuley, A., Stewart B., Siemens G., Cormier D. (2010). Massive Open Online Courses - Digital ways of knowing and learning. The MOOC Model for Digital Practice. George Siemens and Dave Cormier.
- Neto, G. G. C., Gomes, A. S., Tedesco, P. (2003). Elicitação de Requisitos de Sistemas Colaborativos de Aprendizagem Centrada na Atividade de Grupo, In XIV Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE), vol. 1, n 1, pp. 317–326.
- Sabbatini, R. M., (2007) Ambiente de ensino e aprendizagem via internet: a plataforma Moodle. Instituto EduMed, v.25. Available in: <<http://www.ead.edumed.org.br/file.php/1/PlataformaMoodle.pdf>>. Access in: 03 apr. 2018.
- Siemens, G. (2005) Connectivism: A Learning Theory for the Digital Age. Int. J. Instr. Technol. Distance Learn., vol. 1, p. 1–8.
- Sommerville, I. (2007). Software Engineering – Eight Edition. Harlow: Pearson Education Limited.