colMOOC™

D4.5
2nd evaluation report

Co-funded by the
Erasmus+ Programme
of the European Union

Co-funded by the Erasmus+ Programme of the European Union

The colMOOC: Integrating Conversational Agents and Learning Analytics in MOOCs

D4.5 – 2nd evaluation report

Project number:	588438-EPP-1-2017-1-EL-EPPKA2-KA
Grant Agreement No:	2017-2841/001-001 (Amendments 002 & 003)
Project acronym:	colMOOC
Project title:	The colMOOC: Integrating Conversational Agents and Learning Analytics in MOOCs
Programme, Key action, Action type, Topic:	E+ KA2: Cooperation for innovation and the exchange of good practices, Knowledge Alliances
Start date of the project:	01/01/2018
Duration:	39 months (extended: 03/31/2021)
Project web site:	https://colmooc.eu/

Deliverable type:	Report
Deliverable reference number:	D4.5
Deliverable title:	2nd evaluation report
WP contributing to the deliverable:	WP4
Delivery date:	02/28/2020 (M26)

WP Leader:	CERTH
Responsible organization:	CERTH
Abstract:	This document describes the technical components and the technical evaluation of the second version of the integrated colMOOC platform implemented within Work Package 4 “System development, integration and evaluation”. It provides an overview of the integration status of the different components, the hosting infrastructure and the improvements that have been performed since the latest version.
Keywords:	Technological roadmap, development, components

Dissemination level:	Public

Authors and Reviewers[1]

Efstathios Nikolaidis, Centre for Research and Technology Hellas (CERTH), Greece, stathis.nikolaidis@iti.gr

Giorgos Prountzos, Centre for Research and Technology Hellas (CERTH), Greece, gprountzos@iti.gr

REVIEWERS

Santi Caballe Llobet, Universitat Oberta de Catalunya (UOC), Spain, scaballe@uoc.edu

Allison Kolling, Saarland University (SU), Germany, a.kolling@edutech.uni-saarland.de

[1] Name Surname, Organization Name (Organization Name Acronym), Country, e-mail address

Document Change Log

Version	Date (mm/dd/yyyy)	Author (s)	Sections Changed
0.1	12/01/2019	See author list	ALL
0.2	01/30/2020	See author list	ALL
1.0	02/28/2020	See author list	ALL

Executive Summary

The main goal of this document is to present the technical assessment of the 2nd version of the colMOOC platform through performance benchmarking experiments performed on the system's components in order to verify the correct operation and interoperability of the different integrated modules to meet the requirements of the use cases for the end users.

The development status of each component is updated and summarized along with the integration approach and infrastructure that was used towards the implementation of the platform.

Technically, all modules have been deployed at a basic level fulfilling the requirements. The general assessment from this deliverable is that the evaluation showed positive feedback and clearly pointed out the planning path towards the final colMOOC system.

Table of Contents

List of Figures

List of Acronyms

Acronym	Description
AB	Advisory Board
DLP	Deliverable Lead Partner
EB	Ethics Board
IPG	Intellectual Property Management Group
PC	Project Coordinator
QA	Quality Assurance
QC	Quality Control
QEG	Quality Evaluation Group
SB	Project Supervisory Board
SM	Scientific Manager
TM	Technical Manager
ToC	Table of Contents
UG	User Group
WPL	Work Package Leader
MOOC	Massive Open Online Courses
CA	Conversational Agent
LA	Learning Analytics
URL	Uniform Resource Locator
API	Application Programming Interface
QA	Quality Assurance
REST	Representational State Transfer
HTTP	Hypertext Transfer Protocol

1 Introduction

1.1 Purpose of this document

In this document we provide the output of creating the integration, testing and technical evaluation of the 2nd version of colMOOC platform. For the fulfilment of the integration plan, partners collaborated on many different occasions, especially for the definition of the necessary integration points between components, but also to establish a common approach regarding testing and evaluation with the main goal to cover the functionality defined by the end-users. The integration of this 2nd version is a continuation of the plan defined in D4.4 "Integrated colMOOC platform 2nd version" and the Architectural Design, continued with the specification of the mechanisms described in D4.1. The goal of revising the integration plan is to support the development by guiding the integration of the software components to the final version of the colMOOC platform, with the agreement on identified interfaces.

Overall, this report shows that the colMOOC project has achieved its objectives so far and fulfilled its key requirements. In addition, the colMOOC technical evaluation activities have allowed for identifying and quantifying the strengths as well as some weaknesses of the platform and planning the path for the final version of the colMOOC platform.

1.2 Integration Overview and current status

In D4.1 (June, 2018), a technological roadmap was established. This roadmap determines the main attributes and timelines for the development of the different components of the colMOOC platform and describes an iterative approach from the initial operation prototype towards the final version of the colMOOC platform. The "walking skeleton" for this technical roadmap is presented below:

During the second year of the project the main goal was to provide a second integrated version of the colMOOC platform which could allow a fast and easy integration framework including a fully successful process of colMOOC activity cycle using real services from WP2-WP6, and services with basic functionality on them.

In this third step of implementation and according to the expected timeline for MS4, a second milestone of a functional second prototype is presented, followed by improvement of the functionalities and using real services from WP2-WP6.

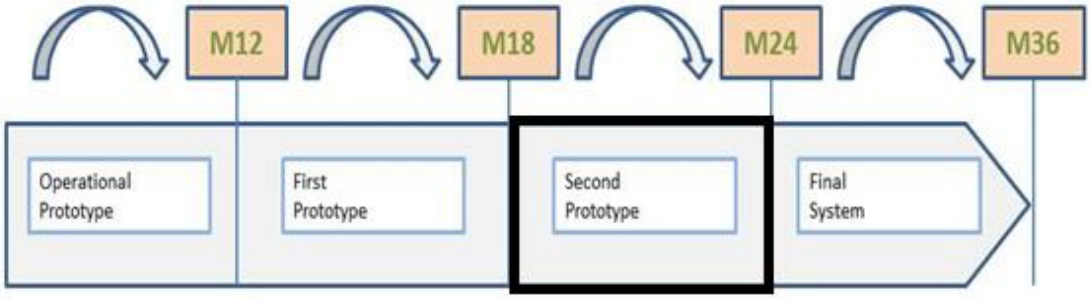

Figure 1: colMOOC walking skeleton

1.3 Second Prototype Architecture

The forming architecture is formulated with the ultimate goal of providing accuracy and functionality for complex collaborative learning scenarios supported by a conversational agent and integrated in MOOC platforms.

The architecture is roughly made up of the following layers:

1. Ingestion layer, containing mechanisms and channels through which data is brought into the platform;
2. Internal services, generic data repositories and communication services being used by the different components;
3. Business layer, containing the components that perform the actual platform-specific capabilities;
4. External facing layer, including the CA, LA modules, interacting with people and entities outside the platform (end-users of the platform).

The first step in a generic flow of the platform consists of new data being pushed into the platform. The new data can originate from specific MOOCs. Note that these MOOCs may act bi-directionally, as they can serve to communicate back with the users upon need. Moreover, incoming data to the platform can originate from other sources as well, such as stored agents.

Once a new piece of data (e.g., student ID in the MOOC) is successfully ingested into the system, the data is stored in a temporary raw data store and the availability of the new piece of data is broadcasted to all interested components.

All interested parties receive the information, which includes a pointer enabling them access to the data, and perform their specific analysis on the new data. Note that the result of such analysis may in turn create a new piece of data which can be of interest to other components (such as LA); this new piece of data will be, once again, available and accessible in the same manner explained above, providing services for analysis.

The final results of the analysis can be added to the database, and, if required, the MOOC platform will be notified in order to identify a new state.

1.3.1 Platform components

Ingestion layer – Serves as the input mechanism into the platform. Different kinds of information can serve as input to the system. Typically, once a new piece of data has been ingested into the platform, it is stored temporarily in a raw storage system, and a proper notification is sent via the service bus to the interested parties.

Internal services – These services are used internally for the proper functioning of the capabilities provided by the various components. These services are also used mostly for data storage and communication. Some generic data analytics and processing services may be used as well.

Business layer – This layer encompasses the components that perform the actual platform specific capabilities. The bulk of the platform is concentrated in this layer, which interacts in turn with all additional layers.

A particular group of components in this layer tackles the analysis of different kinds of data flowing into the platform. Among this group we can find the following components:

- Conversational Agent editor
- MOOC authoring

A second group of components deal more specifically with the analysis. Among these components, there are:

- Learning analytics
- Communication with the MOOCs

External layer – This layer handles the interaction of the platform with external entities, both as input providers and output recipients. There are two main groups of components making up this layer, namely:

- Conversational Agent player
- MOOC student environment

The REST architectural pattern will be used to model the services used by the different services. REST proposes a very lightweight, HTTP-based method of stateless operations between resources in the Web.

2 Technical Evaluation of the system

Following the guidelines of the colMOOC technical evaluation strategy, technical meetings were regularly organized and several test cases were performed so as to validate and verify the 2nd version's achievements. The evaluation focused on covering as many Key Factors as possible with respect to the technical work.

2.1.1 Testing and Evaluation Methodology

Prototyping was used for software development, regression verification, and in-system validation. In other words, it was useful to test whether each component involved in creating and delivering the system worked well with other components, and whether the entire scale of operations ran smoothly.

In order to reach conclusions regarding the technical evaluation, it is noteworthy to refer to the extended efforts of testing that we committed upon through teleconferences with other partners, in order to solve both intra-communication problems with other components, as well as internal functionality misbehaviours.

At the project integration level, we used weekly plenary meetings to provide updates on the current status of the different parts of the project in order to accomplish specific deadlines following a SCRUM testing methodology [1] [2].

The following paragraphs describes this testing methodology, which was tailored specifically for the integration of the 2nd version of the colMOOC platform. Our testing framework was designed in reference to the requirements. As it has been previously mentioned, colMOOC is a use-case-driven project with three different use cases to validate the under-deployment platform. These use cases consisted of a sequence of steps that describe the interactions between the users and the system.

Within colMOOC, the testing framework meetings were organized and facilitated by the scientific manager assisted by the technical manager of the project. Every week, a different sequence of the storyline from the Use-Cases was set by the scientific manager to be tested, creating a backlog – a list of tasks to perform during the meeting. These tasks were forwarded before meetings via email to all participants and were populated on colMOOC's DokuWiki under the page "Technical Telcos – Minutes and action points". Through this process, and assigning roles to the participant "actors", the testing procedure was well tuned and all members had a clear idea about their responsibilities.

In this integration of the 2nd prototype of the platform all testing scenarios consisted of two main phases. The phase of the design-editor and the phase of the activity-player. The main "acting roles" in the sequence of testing scenarios were those of the teachers and students. These roles are incarnated in the context of the External facing layer of the platform.

The aforementioned tools were serving as the principal criterion for the stakeholders to monitor and to evaluate the process. Furthermore, an additional tool serving for internal purposes as a message logger was used for the graphical representation and monitoring of the messages exchanged between the different components.

The main idea of integration testing was to combine and test the interface between the key components gradually, and eventually to expand this combination in order to test all the integrated components of the platform fulfilling a complete flow that describes a Pilot Use Case scenario.

Thus, the scientific manager, orchestrating the whole process and with respect to the predefined testing plan, was requesting for actions from technical partners.

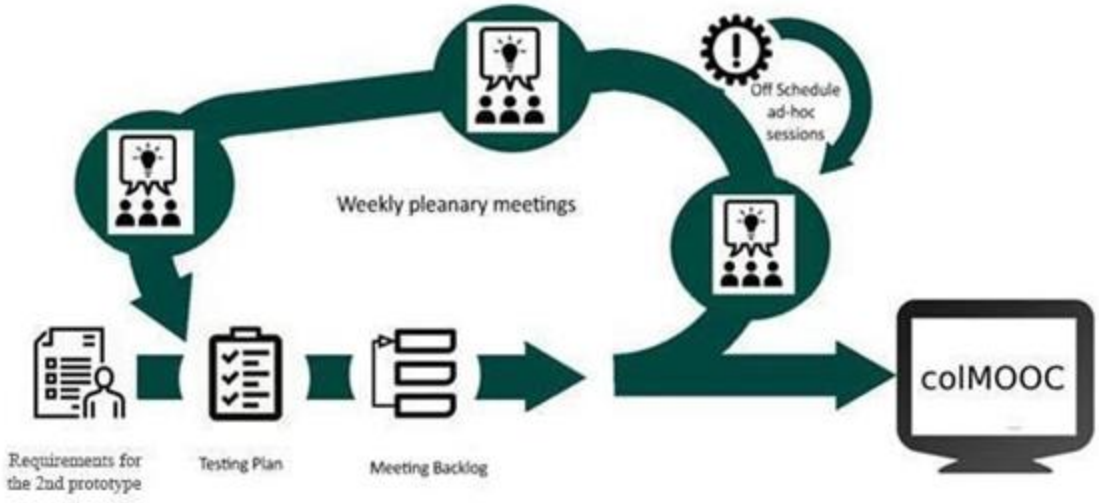

Figure 2: Testing and Evaluation Methodology

This approach emphasizes the value of collaboration and the rapid feedback that iterative usability evaluation can provide, where its effectiveness typically relies more on the involvement of End Users - usability experts.

To summarize, plenary meetings were serving as weekly reviews for testing the platform, during which the teams demonstrated the new functionality to the end-users-experts who in turn provided feedback that could, in turn, influence the next meeting. Technical overview, combined with the feedback from the stakeholders, resulted in changes to the delivered functionality, but also in revising or adding items to the product backlog. Identified issues, change requests, and new tasks were accumulated after the sessions to feed the Issue Management tool that is described in the following subsection.

Various ad-hoc sessions were also scheduled jointly with the regular meetings. The main goal of those sessions was to fill the software gaps between components.

For inter-task work, a more informal communication protocol was used via direct mailing or Skype group chat/calls.

2.1.2 Issues Management

There is no doubt that issue tracker is one of the key factors for evaluating the software process development. It provides a platform to support software development and maintenance activities such as issue reporting, tracking and resolution. Issue tracking (such as a bug report or a feature enhancement request) is a social and collaborative process in which an issue tracker serves as a communication hub and channel between the developers and the QA (Quality Assurance) [3] [4] team.

For the issues found during development and technical tests of the colMOOC platform, the consortium partners used GitLab [5] embedded issues management approach to store issues, create milestones and distribute the technical work needed to overcome these issues.

GitLab issue tracker serves the following purposes:

- Safe and reliable method for the team to raise issues.
- Track and assign responsibility to specific people for each issue.
- Analyze and prioritize issues easily.
- Record issue resolution for future reference and project learning.
- Monitor overall project health and status.

For the better organization of the issues, specific labels were created and used in order to prioritize and map the issues with the platform components, as depicted in Figure 3.

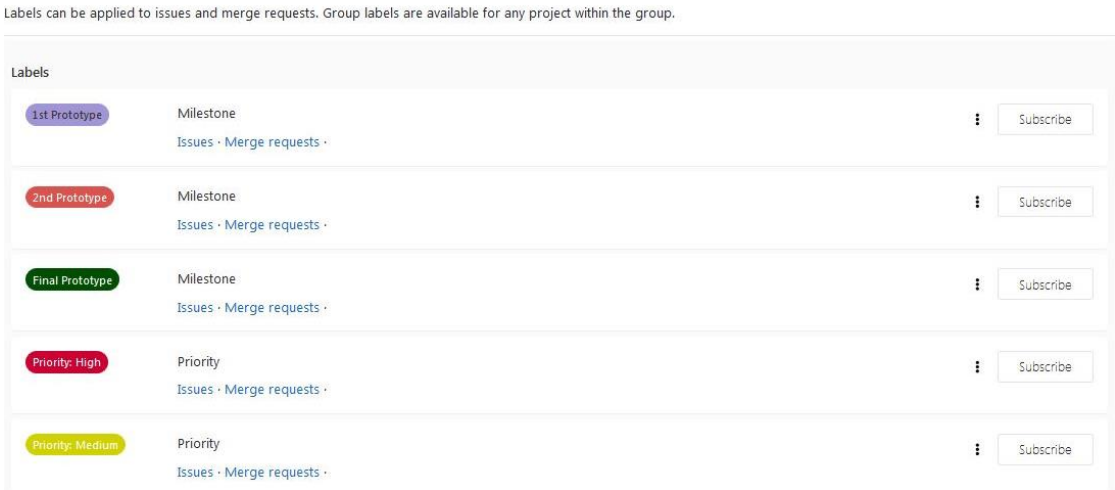

Figure 3: Issues Management for colMOOC components - Labels

Finally, in order to impose the time plan for addressing the reported issues, specific Milestones were used with reference to the technical weekly meetings The management of issues, labels and milestones was open to all technical partners and was done at a repository level, and thus changes were made easily, when needed. A component that showed any existing issue could be aggregated at the root level colMOOC-project in GitLab; this made it easy to have an overview of the issues of the whole framework.

2.1.3 Evaluation of Key Factors

For the evaluation of the quality of the 2nd version of the platform we chose to present some quality metrics of the modules that are considered key factors of the implementation.

There are many scalability dimensions in the integration and mechanisms used which affect favorably the overall performance and throughput of the system. Currently, the deployed system can comfortably accommodate the anticipated load of the colMOOC pilots, and has the capacity to support a higher load, given the current installation and deployment. In addition, there are various scalability factors, affecting performance that can be applied when the system load becomes considerably larger.

- **Number of API Calls**
  For scalability and fault tolerance, the API can run with a number of servers acting as cooperating message brokers, in which cooperation is done in order to provide continuous service. Running multiple brokers means that, for each partition, there shall be a single broker acting as the designated leader and a list of brokers acting as replicas. Currently, colMOOC's API is deployed over 5 brokers. The number of brokers can be scaled up based

on need, but for the foreseeable future there is no expectation that the platform would require more brokers to be deployed.

- **Number of Students and groups**

For the evaluation of the quality of the 2nd version of the platform we had 1,380 Students running the colMOOC Player module from three different external platforms. The students participated in 13 different activities as random pairs of two depending on the platform they had previously enrolled. Those 2,392 different groups of Students ran the colMOOC Player in sessions with a large variety of Agents using the colMOOC platform simultaneously.

- **Number of Chat posts**

By running a total number of 2,392 different sessions, there were 114,684 chat posts (average number of 47.94 chat posts per session, and 23.97 chat posts per student per session).

- **Number of Agent interventions**

For the evaluation of the Agent's performance, we used different Agents from various created MOOCs. The Agents produced 32,531 interventions (average number of 13.59 interventions per session).

When increasing the total load to the upper limits documented above while running MOOCs with users from different platforms, there were no significant differences regarding the overall response of the system and all individual modules (colMOOC Editor, Player or Learning Analytics). The colMOOC platform was running without any connection issues and managed to deliver for what it was designed for.

3 Summary

This deliverable reports on the integration of the second prototype of the colMOOC platform. This document presents the development status of each component, the integration approach, and the methodology for evaluation and validation of the platform. The deliverable also presents the operational use case scenarios that were implemented in order to validate the functionality of the platform and the different modules.

The second prototype of the system will be tested in various pilot use cases and the evaluations will help to develop the final version of the system with enhanced capabilities and more features.

References

[1] Mike Cohn, "Succeeding with Agile: Software Development Using Scrum", Addison-Wesley, 2010

[2] Schwaber, Ken; Sutherland, Jeff (November 2017), "The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game", retrieved May 13, 2020

[3] Bourque, Pierre; Fairley, Richard E., eds. (2014). Guide to the Software Engineering Body of Knowledge. IEEE Computer Society. ISBN 978-0-7695-5166-1.

[4] Kelemen, Zádor Dániel; Kusters, Rob; Trienekens, Jos (December 2012). "Identifying criteria for multimodel software process improvement solutions - based on a review of current problems and initiatives". Journal of Software: Evolution and Process. 24 (8): 895–909. doi:10.1002/smr.549. S2CID 14382496.

[5] GitLab Inc. GitLab DevOps tool, version 13.10.2, Available at https://about.gitlab.com/