MWDEX: A Moodle Workshop Data EXtractor

Emiliano Acquila-Natale
Organization Engineering, Business Administration and Statistics
Universidad Politécnica de Madrid
Madrid, Spain
emiliano.acquila@upm.es

Santiago Iglesias-Pradas
Organization Engineering, Business Administration and Statistics
Universidad Politécnica de Madrid
Madrid, Spain
s.iglesias@upm.es

Ángel Hernández-García
Organization Engineering, Business Administration and Statistics
Universidad Politécnica de Madrid
Madrid, Spain
angel.hernandez@upm.es

Julián Chaparro-Peláez
Organization Engineering, Business Administration and Statistics
Universidad Politécnica de Madrid
Madrid, Spain
julian.chaparro@upm.es

Isabel Rodríguez-Ruiz
Organization Engineering, Business Administration and Statistics
Universidad Politécnica de Madrid
Madrid, Spain
isabel.roruiz@alumnos.upm.es

ABSTRACT

Research on learning analytics has primarily focused on certain stages of the learning analytics cycle, such as analysis techniques and data visualization. However, there is still an alarming scarcity of available tools for data collection and preparation. This study presents a software application that facilitates the extraction and subsequent analysis of peer assessment-related data from Moodle Workshop activities: Moodle Workshop Data Extractor (MWDEX). The application may be run locally or remotely and uses the Moodle web services layer to access data in a secure manner. The study provides an overview of the design and main components of the system and describes how to configure and run MWDEX with an illustrated example.

CCS CONCEPTS

• Applied computing~Learning management systems • Information systems~RESTful web services • Applied computing~Computer-assisted instruction • Applied computing~Collaborative learning • Applied computing~Computer-managed instruction • Applied computing~E-learning

KEYWORDS

Moodle, workshop, peer assessment, data preparation, learning analytics.

TEEM'19, October 16–18, 2019, León, Spain

ACM ISBN 978-1-4503-7191-9/19/10…$15.00
https://doi.org/10.1145/3362789.3362899

ACM Reference format:

Acquila-Natale, E., Hernández-García, Á., Iglesias-Pradas, S., Chaparro-Peláez, J., & Rodríguez-Ruiz, I. 2019. MWDEX: A Moodle Workshop Data EXtractor. In *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2019) (León, Spain, October 16-18, 2019)*, *ACM, New York, NY, USA*, 7 pages. https://doi.org/10.1145/3362789.3362899

1 Introduction and problem statement

This introductory section aims to provide the reader an outline of the context, main motivations and objectives of the study. The first subsection situates the research in the emerging field of learning analytics, focusing on one of the least explored stages of the learning analytics cycle: data preparation. Then, the second subsection presents the learning context, peer assessment, which is then followed by a summary of the different ways to implement peer assessment in the most popular learning management system, Moodle, with emphasis on the Moodle Workshop module and its advantages and current limitations. To overcome some of the shortcomings of the Moodle Workshop module, and more specifically those related to peer assessment-related data preparation for any purpose, Section 2 presents an application for extraction of such data (Moodle Workshop Data Extractor, MWDEX); the subsections in Section 2 explain in detail the design and implementation of MWDEX, and illustrate its use with an example. Finally, Section 3 summarizes the main conclusions and future lines of research.

1.1 Learning analytics and data preparation

The field of learning analytics has experienced great advances in the past years, led by the promise of improvement of teaching and learning processes. Learning analytics focuses on the extraction, analysis and interpretation or understanding of the digital traces left by all learning agents. Its most usual practical application falls in the realm of IT-supported learning, e.g. flipped classroom [1], blended learning [2–4], traditional online distance learning [5, 6], mobile learning [7], MOOCs [8], serious games [9], simulation-based learning [10], etc., using log data from users' interactions in the virtual environment. However, the sources of data generation are not limited to interactions in digital spaces, but also to data captured in physical spaces; e.g. video captured during classroom activities, movement information from body sensors or eye-tracking devices [11], etc.

Nonetheless, activity of teachers or learners and data collection are only the first step in the cycle of learning analytics processes [12–15], followed by data processing and analysis that serve as basis for potential interventions. According to [16], educational data are seen as a proxy between social-cognitive processes and their outputs in learning analytics processes, where the outputs must be explained in terms of the results of the analysis. The authors identify four different types of data—primary data, data resulting from measurements of artifacts, repurposed data and transformed data—, but they identify two different routes to educational intervention and outcome: direct analysis, using the first three types of data, and data transformation and analysis, using transformed data.

This subtle distinction is particularly interesting to address, as most current research on learning analytics generally focuses on methods for data collection and development of different data mining techniques and algorithms, as well as the results of the application of such analysis techniques and the visual presentation of data. However, little attention is put into data pre-processing, which is an essential part of the learning analytics cycle. Even when the methods for data collection are reported and well documented, most often we find that researchers then turn their attention to the analysis of such data and rarely report how exactly are data being prepared and aggregated. As a result, knowledge about available tools to prepare data for learning analytics purposes is limited.

1.2 Peer assessment

Peer assessment refers to the students' evaluation of the work done by their classmates, using a set of assessment criteria determined by the instructor, most generally in the form of a rubric. Peer assessment is generally considered an adequate assessment method in cooperative and collaborative learning settings [17, 18]. This type of assessment facilitates the development of critical thinking of students and fosters self-reflection about their own work. In team-based project, peer assessment may be used in the evaluation of students by the rest of the members of the team, as well as for the assessment of the learning outcomes of a team as a whole by the rest of the team. In addition, peer assessment may also include self-assessment to reinforce the critical evaluation of the student's own work [19].

The benefits of this kind of assessment for students, among others, are generally associated with deeper learning experiences through reflection and self-criticism, as well as higher student involvement with the course and with the team—as a consequence of their inclusion as active part in the assessment and grading process—, a better and deeper understanding of assessment criteria and a higher perception of fairness through student empowerment [20, 21]. For the instructor, the main benefits refer to the improvement of the overall learning experience for students, but also to saving time.

1.3 Peer assessment in learning management systems: Moodle and Moodle Workshops

Moodle is an open-source learning management system, written in PHP, which supports the creation of online learning communities in different teaching and learning delivery modes—blended learning, online learning, flipped classroom, etc. Moodle is the most used learning platform in the world, with more than 19 million registered courses, over 162 million users and over 106,000 sites in 228 countries [22]. Moodle provides users with a set of tools centered on the student and collaborative learning by facilitating the interaction between instructors and students.

There are currently three main activity modules in Moodle that support peer-assessment: workshops, message boards (forum) and database. Each of these modules have a series of advantages and disadvantages [23], but most often instructors implement their peer-assessed activities as workshops. Moodle Workshops offer by and large the largest set of functionalities to support peer-assessment activities.

In Moodle workshops, students complete and submit their assignments following a series of five phases planned by the instructor: setup, submission, assessment, grading/evaluation and closing of the activity. The progress of the activity is visible in the Workshop Planner Tool, along with all the tasks that the user needs to perform during that phase and the duration of the phase. The idea is that students submit their own assignments and then receive a predefined number of submissions from other students that they must assess, anonymously or not, according to the instructions and assessment criteria set by the instructor; if the instructor requests self-assessment, students must assess their own submissions too.

During the setup phase, instructors may modify all the workshop settings: dates, grading strategy, information, assessment rubrics, etc. Students cannot perform any activity while on the setup phase. In the submission phase, students are now allowed to submit their assignment. The instructor may define the start and end dates of submission, as well as check the current status of submission of all students. The assessment phase is when students perform and submit their assessment of other students' submissions—and of their own submission, if self-assessment is enabled—until the specified date and time. Depending on the setting, students must submit a final grade or grade different aspects of the assignment based on the specified assessment rubrics, and they also may provide a general feedback or specific feedback about the different aspects. Assessment is followed by grading, which involves calculation of the final grade and provision of feedback for students, both authors and reviewers. No further modifications to the submissions are allowed in this phase. Instructors may also add their own grading and publish the results. Closing means that the grades are written to Moodle's Gradebook and made visible to students. Students may also view their submissions and assessments.

An interesting feature of Moodle Workshop is that, in addition to the submission grade—the one given by one student to another—, Moodle also calculates an assessment score, which indicates how close the grade given by a student is to the average grade of the assessed student. This gives an idea about how 'good' or reliable a student is at assessing the work of their peers; Moodle allows different ways to calculate assessment scores.

1.4 Problem statement

The most important drawback of Moodle Workshops currently is the inability to export detailed data about the different assessments performed by students. While instructors are able to export final grades from Moodle gradebook or explore the information related with any assessment in their browser (see Figure 1 and Figure 2), data about individual assessment, partial grades (e.g. grades in specific aspects of a rubric) or written feedback cannot be exported at this moment. Such exported data might be valuable to create personalized reports using assessment rubrics or to perform different types of analysis of peer-assessed activities.

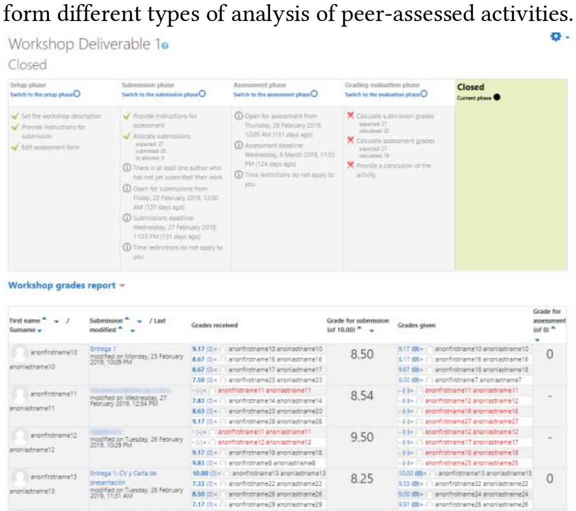

Figure 1: Overview of Moodle Workshop for the course instructor, including assignments and peer-assessment submissions.

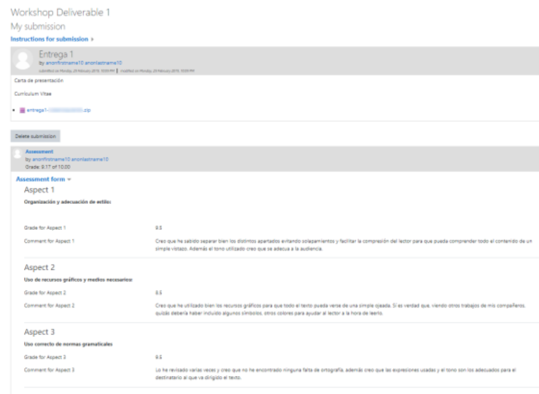

Figure 2: Detail of a graded submission using an assessment rubric with three aspects and feedback.

There are two possible solutions to this problem. The first involves creating a tailored SQL (Structured Query Language) query to the database and later processing of the information from the different tables, but it requires the use of custom reporting plugins (e.g. https://docs.moodle.org/37/en/Custom_SQL_queries_report), as well as the necessary privileges, knowledge about SQL syntax and the structure of Moodle database. Therefore, this solution might be considered only temporary and highly dependent on the specific activity of interest, because any slight change would entail adaptation of the SQL query.

The second way to address this problem is the one proposed in this study. In the vein of [24, 25], the solution involves developing an intermediate application that, using Moodle web services, facilitates access to workshop-related data using a simple and intuitive interface. In addition, such a solution also aims to contribute to practice on learning analytics by providing a helpful tool for data preparation. The following section details the implementation of such application, named Moodle Workshop Data Extractor (MWDEX).

2. MWDEX: Moodle Workshop Data EXtractor

The main objective of MWDEX[1] is to extract data related to workshop activities from Moodle and return a dataset that the instructor may analyze, edit and manipulate. The dataset must include all the available and relevant information about peer-assessment of the submissions: who submitted the assignment, who performed the assessment, the final grade and global feedback, specific grades and feedback of any given number of aspects of an assessment rubric, and whether the grading corresponds to peer- or self-assessment.

In order to make it independent of LMS changes, MWDEX will be designed as an external application that will extract Moodle workshop data using Moodle web services. This ensures that the structure and order of the LMS architecture is preserved. The ap-

[1] The source code may be accessed and downloaded at: https://github.com/TIGE-UPM/MWDEX. MWDEX is licensed under a MIT License.

plication must retrieve all assessment-related information available, transform it and return a file that is human-readable, easy-to-manipulate and edit, and with analysis capabilities (e.g. MS Excel); alternatively, the output may also be easy to process by any other program for extended functionality (e.g. JSON).

Using web services to perform the queries requires the use of HTTP requests to Moodle, otherwise data extraction would not be possible. Because data is transmitted in JSON, it is necessary that the application can handle JSON. There are three additional desirable characteristics in the final system: 1) compatibility with different versions of the LMS; 2) client platform independence, so that it may be used in different operating systems; and 3) high processing speed, for increased performance.

2.1 Overview of MWDEX

Upon the project design explained in the previous section and summarized in Figure 3, it is necessary to describe the architecture of MWDEX, which consists of two main components: the Moodle web service and the web application.

The web service enables access to some Moodle pre-installed functions (detailed in section 2.2) that establish a communication using the REST protocol with the web application. The web application makes the data request to the LMS; this request is handled via Moodle web services layer, which gives a response with the necessary data using JSON. Upon data reception, the web application processes the data and generates the resulting MS Excel file or JSON file with the information about each peer assessment.

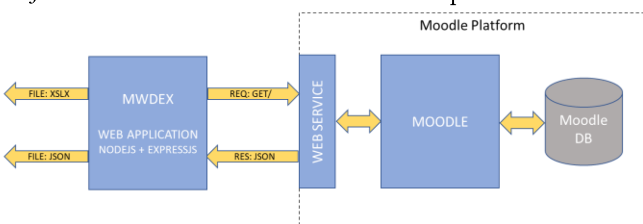

Figure 3: Graphic overview of MWDEX.

The process depicted in Figure 3, which summarizes MWDEX operation, can be explained in the following sequential stages:

1. The user (most likely, a teacher or course administrator) accesses the web application (leftmost part of Figure 3), and enters his or her authentication details in Moodle, including URL of Moodle and name of the web service. The web service needs to be activated in Moodle; the different functions of the web service are explained in further detail in section 2.2.
2. The web application requests the user's token.
3. If authentication credentials are correct, Moodle returns the token that the system will use for the following petitions; else, MWDEX shows a logon error message.
4. Using the token, the web application requests the list of courses available for that user in Moodle, only showing those in which the user is enrolled in.
5. Moodle returns the list of courses in JSON format. The list may be empty if the user cannot access any course or no course exists in the Moodle LMS.
6. The web application shows the list of courses for the user to choose one.
7. The user selects a course and sends a request for the workshops available in that course.
8. Moodle returns the list of workshops for that course in JSON format. The list may be empty if there no workshop exists in the Moodle LMS.
9. The web application requests the list of users enrolled in the course and the list of course workshops.
10. Upon user confirmation, the web application requests all the information of assessments for the selected workshop.
11. Moodle returns workshop grading information in JSON format.
12. For each user, the web application populates a variable listing all the information about each submission assessed. This process is explained in more detail later on in this document.
13. After collection of all data, the web application generates the output file (in MS Excel format or JSON).

2.2 Implementation of the web service

This section explains how to set up the web service in Moodle, in order to have MWDEX access the necessary information to export peer-assessment-related data. Even though web service activation in Moodle is a straightforward process, it is omitted in this document due to length limitation. We refer the reader to the following URL for further information: https://docs.moodle.org/37/en/Using_web_services). Because all the required functions are provided natively by Moodle, there is no need to develop any external plug-in to access data. A big advantage of using standard functions instead of a plug-in is that it makes the code easier to maintain and independent of changes in Moodle, unless the standard functions are changed.

In order to access the workshop-related data from Moodle, it is necessary to enable the web service and activate the REST protocol. These options appear in Moodle's administration menu, under the section "External services". In order to create the web service, it is necessary to create a new service with the following functions:

- *core_course_get_courses*: Return course details. Required capabilities: moodle/course:view, moodle/course:update, moodle/course:viewhiddencourses
- *core_enrol_get_enrolled_users*: Get enrolled users by course id. Required capabilities: moodle/user:viewdetails, moodle/user:viewhiddendetails, moodle/course:useremail, moodle/user:update, moodle/site:accessallgroups
- *mod_workshop_get_assessment*: Retrieves the given assessment. Required capabilities: none.
- *mod_workshop_get_assessment_form_definition:* Retrieves the assessment form definition. Required capabilities: none.
- *mod_workshop_get_grades_report*: Retrieves the assessment grades report. Required capabilities: none.
- *mod_workshop_get_workshops_by_courses*: Returns a list of workshops in a provided list of courses; if no list is provided all workshops that the user can view will be returned. Required capabilities: mod/workshop:view

The above operation activates the web service, but it is also necessary to configure access authorization. Administrators can grant authorization to individual users for using the web service in the "External services" menu, and they can then generate access tokens—if required—for authorized users in the "Manage tokens" section.

2.3 Front-end: an annotated example

The main component of MWDEX is the web application. Again following [24, 25], it was decided to use Node.js and the Expressjs framework to ensure cross-platform compatibility across operation systems, and to enable local or remote execution of the code. Being an event-oriented programming language that allows a high number of concurrent connections, Node.js offers improved execution speed and performance. The development also includes an additional JavaScript library, 'exceljs/modern.nodejs' to export workshop data to MS Excel.

The following example details the use of MWDEX and how the user, most likely a teacher or administrator, interacts with the application. More particularly, the example focuses on an instructor's use of MWDEX. The use case scenario is as follows:

1. An instructor with a valid user in the Moodle LMS instance accesses MWDEX using his or her web browser at the specified URL, either local or remote. By default, the application is listening on the port 3000. Upon loading of MWDEX in the browser, the interface prompts for logon credentials: username, password, URL of the Moodle LMS and name of the web service (Figure 4).

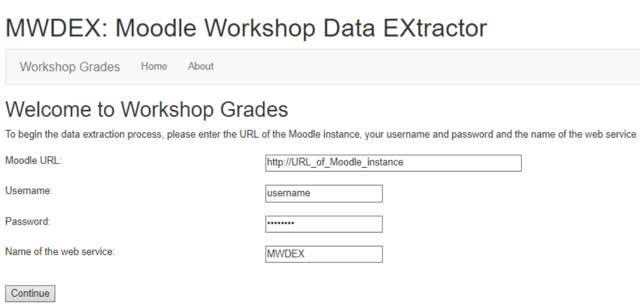

Figure 4: MWDEX home page.

2. After authentication, the web interface of MWDEX displays a list of the available courses. The list only shows those courses in which the instructor is enrolled at that moment (Figure 5).

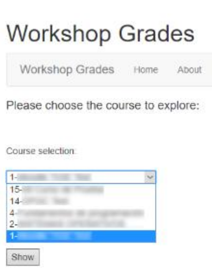

Figure 5: MWDEX screen for course selection.

3. The instructor chooses one course and then MWDEX shows a list of the available workshops in that course (Figure 6).

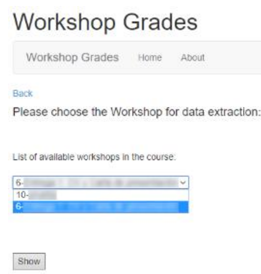

Figure 6: MWDEX screen for workshop selection.

4. When a workshop is selected, the interface shows a list of the students participating in the activity. The user must then choose the output format (.xlsx or .json) and confirm the selection (Figure 7).

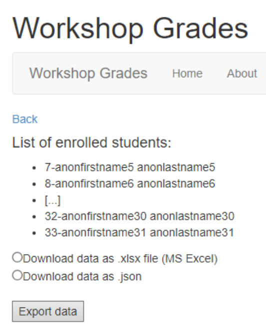

Figure 7: List of enrolled students (edited due to size limitations) and output data format selection.

5. Once all data have been retrieved, it is necessary to transform them so that each output file associates each user's submission with its corresponding grades, reviewers, feedbacks and assessments. This is where the main part of the code of MWDEX gets executed.

At this point, there are two main data structures involved: an array of users, with a list of the students enrolled in the selected course, and a multidimensional array of heterogeneous data. This data array includes four different subarrays: (1) an array with the different aspects of the assessment rubric (i.e. aspect grades and aspect feedback); this array uses an assessment_id as identifier; (2) an array that has information about the global feedback of each assessment, and which uses an assessment_id as identifier; (3) an array of submissions, which has information about submission IDs, assessment IDs and reviewing and reviewed (i.e. grading and graded) student; this array is not used to create the MS Excel file but it is necessary to generate the JSON output file; and (4) a data array including the identifiers of each submission, assessment, users (reviewer and reviewed), as well as final grade and average grade of each aspect of the rubric.

The generation of the Excel file (the generation of the JSON file is similar, albeit with some slight differences; it is not included in this document due to length limitations) is performed by using a function to match all data and a variable to temporarily store all data before writing it to a file; this variable also includes an additional field to differentiate between peer-assessed and self-assessed submissions. The function goes through the data array; for each element, the code makes three different loops: the first traverses the user array and features a nested loop with information about the names of grading and graded student, final grade and average grade of each aspect of the rubric; the second loop populates the result variable with the global feedback of each submitted assessment; finally, the third loop includes incorporates the feedback and grade of each aspect of the rubric to the result variable. The resulting variable has the following form:

- [reviewer, reviewed, final grade, average grade of rubric aspects, global feedback, grade of each aspect, feedback of each aspect, self-/peer-assessment]

This variable is then written to an MS Excel file using the 'exceljs/modern.nodejs' library.

6. Once the application has finished processing the data and created the output file, the user can download it. The resulting file—in this example, a file with extension .xlsx—may be opened using an external application—e.g. MS Excel, OpenOffice Calc, etc. Figure 8 shows the output of the example; in this case, each assignment had a maximum of three peer-reviews and self-assessment, and also included grading of three different aspects of an assessment rubric with feedback in each of the aspects. Blank spaces in grades represent a missing submission of the assessment and blank spaces in feedback represent that no feedback was provided.

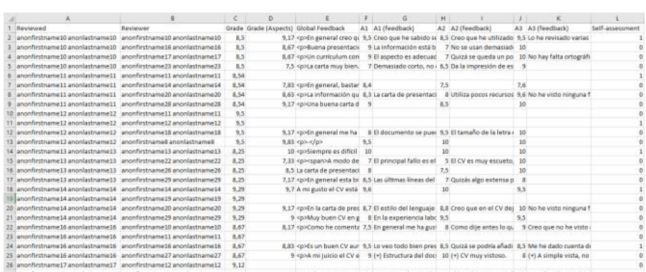

Figure 8: Example of the output MS Excel file returned by MWDEX.

7. With the output file, instructors now have a complete view of all the results of the peer-review process, and it is also possible for them to copy and paste the results into a grading master file, if they have one.

Additionally, and taking advantage of some of the added capabilities of the spreadsheet software, it is also possible to gain insight about the results; for example, the instructor may create graphs and analyze aggregated results using a simple pivot table (Figure 9). While this example shows data from a course with a limited number of students, these operations may also be applied to courses with a very large number of students, where getting insights about the results of the Moodle workshop (as in Figure 1 and Figure 2) might prove very difficult and time-consuming.

Figure 9: Pivot table in MS Excel showing aggregated results of the example MS Excel output file.

3. Conclusion

This study addresses the need for new tools in the emergent field of learning analytics by presenting an application to extract peer assessment-related data in Moodle Workshops: MWDEX. The application, which may run locally or remotely, uses Moodle web services to access, extract, process and export all data related to the author, the reviewer, global grades and feedback, and individual grades and feedback of an assessment rubric. This document explains the software design process and the technical and architectural principles for the development of such tool.

We believe that the implementation of MWDEX meets the needs of instructors who require access to all peer assessment-related information for basic analysis and processing, but beyond that it can also be helpful for data preparation in learning analytics. More particularly, the choice of formats of the output file offers a solution that is extendable, easy-to-integrate and compatible with basic and advanced tools for analysis, as well as other complementary services. Future work includes validation of the tool with instructors in real learning settings.

ACKNOWLEDGMENTS

Research reported in this publication was supported and funded by Universidad Politécnica de Madrid under the *"Ayudas a la Innovación Educativa y a la Mejora de la Calidad de la Enseñanza"* program, with grant number IE1819.0904.

REFERENCES

[1] MN Giannakos, K Chorianopoulos and N Chrisochoides (2015). Making Sense of Video Analytics: Lessons Learned from Clickstream Interactions, Attitudes, and Learning Outcome in a Video-Assisted Course. International Review of Research in Open and Distributed Learning, 16(1), 260–283

[2] S Kotsiantis, N Tselios, A Filippidi and V Komis (2013). Using learning analytics to identify successful learners in a blended learning course. International Journal of Technology Enhanced Learning, 5(2), 133–150.

[3] DT Tempelaar, B Rienties and B Giesbers (2015). In search for the most informative data for feedback generation: Learning analytics in a data-rich context. Computers in Human Behavior, 47, 157–167.

[4] AF Agudo-Peregrina, S Iglesias-Pradas, MÁ Conde-González and Á Hernández-García (2014). Can we predict success from log data in VLEs? Classification of interactions for learning analytics and their relation with performance in VLE-supported F2F and online learning. Computers in Human Behavior, 31, 542–550.

[5] J Ma, X Han, J Yang and J Cheng (2015). Examining the necessary condition for engagement in an online learning environment based on learning analytics approach: The role of the instructor. Internet and Higher Education, 24, 26–34.

[6] Á Hernández-García, I González-González, AI Jiménez-Zarco and J Chaparro-Peláez (2015). Applying social learning analytics to message boards in online distance learning: A case study. Computers in Human Behavior, 47, 68–80.

[7] G Fulantelli, D Taibi and M Arrigo (2015). A framework to support educational decision making in mobile learning. Computers in Human Behavior, 47, 50–59.

[8] M Khalil and M Ebner (2017). Clustering patterns of engagement in Massive Open Online Courses (MOOCs): the use of learning analytics to reveal student categories. Journal of Computing in Higher Education, 29, 114–132.

[9] C Alonso-Fernández, AR Cano, A Calvo-Morata, M Freire, I Martínez-Ortiz and B Fernández-Manjón (2019). Lessons learned applying learning analytics to assess serious games. Computers in Human Behavior, 99, 301–309.

[10] AB Hernández-Lara, A Perera-Lluna and E Serradell-López (2019). Applying learning analytics to students' interaction in business simulation games. The usefulness of learning analytics to know what students really learn. Computers in Human Behavior, 92, 600–612.

[11] X Ochoa (2017). Multimodal Learning Analytics, Handbook of Learning Analytics (1st[t]. Ed.), pp. 129–141. Society for Learning Analytics Research.

[12] D Clow (2012). The Learning Analytics Cycle: Closing the loop effectively. Proceedings of the 2nd International Conference on Learning Analytics & Knowledge (LAK'12), pp. 134–138.

[13] DA Gómez-Aguilar, FJ García-Peñalvo and R Therón (2014). Analítica visual en e-learning. El profesional de la información, 23(3). 236–245.

[14] M Khalil and M Ebner (2015). Learning Analytics: Principles and Constraints. Proceedings of the World Conference on Educational Media and Technology 2015 (EdMedia 2015), pp. 1789–1799.

[15] T Elias (2011). Learning Analytics: Definitions, Processes and Potential.

[16] N Nistor and Á Hernández-García (2018). What types of data are used in learning analytics? An overview of six cases. Computers in Human Behavior, 89, 335–338.

[17] S Casal (2016). Cooperative Assessment for Learning in CLIL Contexts. Estudios sobre Educación, 31, 139–157.

[18] P Chin (2016). Peer assessment. New Directions in the Teaching of Physical Sciences, 3, 13–18.

[19] MS Ibarra-Sáiz, G Rodríguez-Gómez and MÁ Gómez Ruiz. Benefits of Peer Assessment and Strategies for Its Practice at University. Revista de Educación, 359. 206–231.

[20] NAE van Gennip, MSR Segers and HH Tillema (2009). Peer assessment for learning from a social perspective: The influence of interpersonal variables and structural features. Educational Research Review, 4, 41–54.

[21] C Adachi, J Hong-Meng Tai and P Dawson (2018): Academics' perceptions of the benefits and challenges of self and peer assessment in higher education. Assessment & Evaluation in Higher Education, 43(2), 294–306.

[22] Moodle.net (2019). Moodle Statistics. Available online: https://moodle.net/stats

[23] EdTech Team (2016). Peer Assessment Guide. Available online: https://sleguidance.atlassian.net/wiki/spaces/Moodle/pages/65307026/Peer+Assessment+Guide

[24] J Chaparro-Peláez, E Acquila-Natale, S Iglesias-Pradas and I Suárez-Navas (2015). A web services-based application for LMS data extraction and processing for social network analysis. New Information and Communication Technologies for Knowledge Management in Organizations, pp. 110–121. Springer, Cham.

[25] Á Hernández-García and I Suárez-Navas (2017). GraphFES: A Web Service and Application for Moodle Message Board Social Graph Extraction. Big Data and Learning Analytics in Higher Education, pp. 167–194. Springer, Cham.